JOINT INSTITUTE FOR AERONAUTICS AND ACOUSTICS

National Aeronautics and
Space Administration

Ames Research Center

JIAA TR-83

Stanford University

Application of the Vortex Cloud Method to Slender, Flapped Delta Wings

by

S. Oh and D. Tavella

December 1987

JIAA TR-83

Application of the Vortex Cloud Method to Slender, Flapped Delta Wings

by

S. Oh and D. Tavella

December 1987

ABSTRACT

The vortex cloud method is explored as an alternative to more traditional approaches to the study of delta wing aerodynamics. Typically, the aerodynamics of slender delta wings with separated flow is analyzed by either CFD or panel methods. Within limits, the vortex cloud offers definite advantages over such methods. Contrasted with numerical solutions of the Euler or NS equations, the vortex cloud leads to a far better definition of the shear layers characterizing the flow over the leeside of a delta wing, and is also much more economical. The vortex cloud is also more economical than panel methods.

The vortex cloud method consists in simulating a separated two-dimensional flow with a "cloud" of point vortices. This allows for the treatment of a separated flow problem in an inviscid framework, with great advantage in speed of computation. The vortex cloud approach lends itself ideally to two-dimensional problems, and has been so used in the past. This approach is quite successful at computing important quantities of separated flows, such as Reynolds stresses and Strouhal numbers.

The vortex cloud description of a separated flow can be adapted to the analysis of slender delta wings by recognizing that, in a sense, such a wing gives rise to a locally two-dimensional flow field. As a slender delta wing moves at an angle of attack past a stationary observer, each wing cross section is seen by the observer to "plunge" in the fluid medium. This observation leads to the concept of cross-flow plane analysis, which has been used in the past to implement various versions of discrete-vortex methods to study delta wing aerodynamics.

In this work, the vortex cloud method is rigorously adapted to the study of slender, flapped delta wings by casting the three-dimensional equilibrium problem into a time-marching form through an analogy between a 3-D conical and a 2-D self-similar unsteady flow. The range of applicability of the theory is defined, and calculations of load distri-

butions and wake geometries are compared with experiments and with results from other numerical techniques.

Comparison with experimental data shows that when the restrictions of the theory are respected, agreement is quite good. Typical computational times are about 5 minutes on a VAX 780.

TABLE OF CONTENTS

NOMENCLATURE

a	element of geometric influence coefficient matrix
b	vector element in system of equations
C_p	pressure coefficient
c	constant, vortex core, shear layer core
$\mathbf{e}$	unit vector
f	wing cross section shape function,
	normalized vorticity distribution function inside core
g	dispersion of vorticity inside core
H	total head
K	geometric influence coefficient
l	body panel length
M	angular momentum inside core
m	similarity coefficient
$\mathbf{n}$	unit vector normal to wing surface
p	static pressure
R	resultant force, characteristic length
r	radial direction in cross-flow plane
$\bar{r}$	conical coordiante, Prandtl's similarity variable,
	distance normalized with core radius
S	wing shape function
$\mathbf{S}$	slope of body panel
s	distance along wing cross section
T	characteristic time
t	time like variable

U	$\|\mathbf{U}\|$
$\mathbf{U}$	uniform velocity vector
u	axial velocity component
v	magnitude of projection of velocity in cross-flow plane, disturbance velocity
$\mathbf{x}$	coordinate vector
x	chordwise distance from delta wing apex
z	complex representation of cross-flow plane
α	angle of attack
δ	flap deflection angle
ϵ	wing half apex angle
η	spanwise component of bound vortex
Γ	vortex strength
γ	vorticity distribution function inside core, vortex intensity, vorticity vector on wing surface
ν	kinematic viscosity coefficient
Φ	resultant velocity potential
ϕ	disturbance velocity potential
Ψ	stream function
ρ	density, distance inside core
ξ	vorticity component normal to cross-flow plane
σ	vortex core radius
θ	angular coordinate in cross-flow plane, separation angle
ζ	chordwise component of bound vorticity

Subscripts

c	core
F	flap
l	lower part

n	normal component, new
s	separation point
u	upper part
v	vortex position
W	wing
x, y, z	wing reference coordinate system
∞	far field

1. INTRODUCTION

In the past decades numerous studies have been conducted on the wake flow on the leeside of low aspect ratio delta wings.[1-16] Delta wings usually operate with separated vortical flow regions, as shown in Fig. 1. The vortical flow, which enhances the lift on the wing, is difficult to control with conventional control surfaces. The leading edge flap has been suggested as a control device to increase the maneuverability of delta wings. Various shapes of leading edge flaps have been studied in an attempt to optimize their influence on the aerodynamics of delta wings.[17-24] One of the effects of leading edge flaps is to increase the wing lift-to-drag ratio by causing a thrust component of force to act on the flap surface. Leading edge flaps also have the detrimental effect of partially suppressing the vortical flow, thereby reducing the total lift. These two phenomena have been observed experimentally[17,20] and theoretically.[23]

In previous work,[23] the simple Brown-Michael model[3] for the vortical flow was chosen to investigate the qualitative effects of leading edge flaps on the aerodynamics of delta wings. Due to the crudeness of the vortex model only global quantities, such as lift and drag, were evaluated. It is well known that the Brown-Michael model does not yield satisfactory vortex positions, which results in inaccurate pressure distributions. This work, a continuation of that study, applies the vortex cloud method to the analysis of three-dimensional (3-D) conical flows by means of a two-dimensional (2-D) self-similar unsteady flow analogy. It has been shown that with the simple relationship $x = Ut$, where x represents the chordwise direction of delta wings, a 3-D conical flow can be reduced to a 2-D unsteady flow if geometrical and aerodynamical slenderness are satisfied.[5] This procedure leads to an efficient methodology with good definition of shear layers. Another motivation for the use of this approach is the parallel between the physics of planar shear layers and the delta wing rolled-up vortex layer, as suggested in Fig. 3.[25]

The methodology adopted here consists of an adaptation of an intrinsically 2-D approach originally developed to describe separated flows past bluff bodies.[26–36] The 2-D approach consists in tracing the trajectories of a large number of discrete vortices. A problem facing methods of this type is the large velocities induced by neighboring vortices. Some researchers have used finite core sizes to avoid infinite mutually induced velocities,[27] while others redistributed the vortices at each time step so as to avoid a logarithmically infinite velocity.[31] In Spalart's study,[33] the cloud-in-cell technique combined with a discrete vortex method having finite core size was used to make the scheme computationally efficient. This is necessary in most 2-D problems, where the formation of a "vortex jungle"[33] in the wake would, otherwise, cause the computation to become very expensive. In conical flow, on the other hand, the "vortex jungle" would represent the vortex core. Such a representation of a vortex core in a 2-D unsteady flow is unrealistic due to the appearance of high axial velocities, which would violate the aerodynamic slenderness assumption and could not be modeled in a 2-D flow context.[5] For a 2-D unsteady flow, the imposition of a Kutta condition requires particular care, since it arises from the discretization of body geometry and wake. The conventional description for the Kutta condition is that some amount of vorticity is shed into the surrounding fluid at each time step, with strength and position determined by the satisfaction of appropriate conditions at separation points.[37–40] In this study, the quasi-steady Kutta condition is adopted by fixing the separation angle. In this particular case fixing the separation angle has little influence on the whole vortex system. The solution procedure is by successive 2-D computations, where at each time step the cross-section of the wing is modeled with discrete singularities, whose magnitudes result from the solution of a system of linear equations. With this treatment of the Kutta condition, matrix inversions, which requires large computational effort, can be avoided at each iteration by decoupling the Kutta condition form the system of equations.

An alternative implementation of discrete vortex methods consists in seeking the equilibrium positions of the vortices simulating the 3-D separated flow. This is accomplished

by systematically altering their strengths and locations. This approach was first applied to conical flows by Smith.[8] Similar procedure using 3-D panel codes have also been reported.[41,42] Those methods requires a matrix inversion at each iteration step, which tends to make them expensive. In addition, the vortex structure revealed by such methods consists of smooth shear surfaces, while experiments have indicated that structures similar to the ones in 2-D shear layers are also found in the vortical flow of delta wings.[25]

In the present study, the increment of the lift-to-drag ratio, lift reduction and the 2-D shear layer type flow, together with the distortion of the vortex system during flap deflection are investigated. A simple merging scheme for the core region is adopted, whereby the vortices which rotate more than a given angle about the core are allowed to merge together at the centroid of the core and the merging vortices. The three dimensionality of the flow field is implemented through a modification of the Kutta condition, the normalization of the flow quantities, and the way the pressure on the wing surface is calculated. The credibility of the present scheme is tested by comparing shear layer shape, core position and strength, as well as pressure distributions with results from other analyses as well as from experiments.[43] The maximum flap deflection angle for which this methodology remains valid is found by using a simple geometrical relationship. A deflection beyond that limit would cause the leading edge vortex system to move under the wing surface, a situation that cannot be accounted for in this work.

2. MATHEMATICAL PROCEDURE

2.1 Governing Equations

In this section a brief review of mathematical studies of a 3-D conical flow is presented. R.T. Jones[1] first introduced a simple coordinate transformation for a low-aspect ratio wing for supersonic flows. The validity of this transformation in the case of an incompressible flow is studied and the criteria of this transformation are investigated. Mathematical details were derived in previous work.[23] The incompressible, potential 3-D conical flow, corresponding to the flow field illustrated in Fig. 1, can be represented by the following three equations:

Laplace's equation:

$$\frac{1}{\bar{r}}\frac{\partial \phi}{\partial \bar{r}} + (1 + \bar{r}^2)\frac{\partial^2 \phi}{\partial \bar{r}^2} + \frac{1}{\bar{r}^2}\frac{\partial^2 \phi}{\partial \theta^2} = 0 \tag{1}$$

Tangency condition on wing and vortex sheets:

$$\left(U + \phi - \bar{r}\frac{\partial \phi}{\partial \bar{r}}\right) f - \frac{\partial \phi}{\partial \bar{r}} + \frac{f'}{f}\frac{\partial \phi}{\partial \theta} = 0 \tag{2}$$

Pressure continuity across vortex sheets:

$$-\frac{p - p_o}{\rho/2} = 2U\left(\phi - \bar{r}\frac{\partial \phi}{\partial \bar{r}}\right) + \left(\phi - \bar{r}\frac{\partial \phi}{\partial \bar{r}}\right)^2 + \left(\frac{\partial \phi}{\partial \bar{r}}\right)^2 + \frac{1}{\bar{r}^2}\left(\frac{\partial \phi}{\partial \theta}\right)^2 = 0 \tag{3}$$

Eq. (1) is Laplace's equation written in the conical coordinate $\bar{r} = r/x$, Eq. (2) is the condition that the wing surface and the vortex sheet, $S(x, r, \theta) = r - x\, f(\theta)$, must be stream surfaces. Eq. (3) expresses Bernoulli's equation on the vortex sheet, postulating that no pressure difference across the vortex sheet exists.

The corresponding equations for a 2-D, unsteady self-similar flow are

$$\frac{1}{\bar{r}}\frac{\partial \phi}{\partial \bar{r}} + \frac{\partial^2 \phi}{\partial \bar{r}^2} + \frac{1}{\bar{r}^2}\frac{\partial^2 \phi}{\partial \theta^2} = 0 \tag{4}$$

$$m\,f\,-\,\frac{\partial\phi}{\partial\bar{r}}\,+\,\frac{f'}{f}\,\frac{\partial\phi}{\partial\theta}\,=\,0 \tag{5}$$

$$-\,\frac{p}{\rho/2}\,=\,\left(\frac{R}{T}\right)^2\left(\frac{t}{T}\right)^{2(m-1)}\left(2(2m-1)\phi\,-\,2m\,\bar{r}\,\frac{\partial\phi}{\partial\bar{r}}\,+\,\left(\frac{\partial\phi}{\partial\bar{r}}\right)^2\,+\,\frac{1}{\bar{r}^2}\left(\frac{\partial\phi}{\partial\theta}\right)^2\right)\,=\,0 \tag{6}$$

Here $\bar{r}=\left(\frac{r}{R}\right)\Big/\left(\frac{t}{T}\right)^m$ is Prandtl's unsteady similarity variable, and T and R are a characteristic time and length. The wing surface and the vortex sheet geometry are represented by $S(t,r,\theta)\;=\;r\;-\;R\left(\frac{t}{T}\right)^m f(\theta)$. Equations (1), (2) and (3) become identical to (4), (5) and (6) respectively if $m=1$ with the conditions $\bar{r}\ll 1$ and $v_x=\phi-\bar{r}(\partial\phi/\partial\bar{r})\ll 1$, where v_x represents a perturbation on the axial velocity component. Setting $m=1$ also guarantees that the transformation $x=Ut$ holds. The conditions $\bar{r}\ll 1$ and $v_x\ll 1$ represent the geometrical and aerodynamical slenderness respectively.

The simple Galilean tranformation $x\;=\;U\;t$ is applicable to a subsonic conical flow only if aerodynamic and geometric slendernesses are satisfied. This is a necessary condition for a 3-D conical flow to be reducible to a 2-D unsteady self-similar flow. A necessary condition can be derived by a perturbation analysis of Eq. (1). For the delta wing problem at an angle of attack with respect to a free stream velocity, the velocity potential can be assumed to be of the form

$$\Phi' \;= x\;\Phi(\bar{r},\theta) \tag{7a}$$

$$=x\;[U\;\cos\,\alpha\;+\;U\;\bar{r}\,\sin\,\alpha\,\sin\,\theta\;+\;\phi(\bar{r},\theta)] \tag{7b}$$

If the angle of attack α and the wing apex angle ϵ are both of order $\bar{r}$, the second term in the bracket in Eq. (7b), which represents the vertical component of the free stream in the cross-flow plane, is of order ϵ^2. Expanding the velocity potential

$$\Phi\;=\;\Phi_0\;+\;\epsilon^2\Phi_2\;+\cdots \tag{8}$$

the governing equation for Φ_2 becomes Eq. (4) up to order ϵ^2 . If α and ϵ were of different order of magnitude, the velocity potential would have to be expanded as

$$\Phi = \Phi_0 + \alpha\Phi_1 + \epsilon\Phi_2 + \alpha^2\Phi_3 + \epsilon^2\Phi_4 \cdots \tag{9}$$

and the governing equation for Φ_1 would fail to reduce to Eq. (4). Thus, the approach used in this study is valid in a range such that the order of magnitudes of the angle of attack and that of the delta wing apex angle are the same as well as that the flow field must satisfy aerodynamic and geometric slenderness.

2.2 Governing Equations: 2-D Unsteady Flow

As mentioned above, an equivalent 2-D unsteady flow is solved to simulate a 3-D conical flow. In this case, the momentum equation takes on a simple form when expressed in terms of vorticity, since no stretching term is present. The governing equations for a 2-D unsteady, incompressible flow are:

Continuity:

$$\nabla \cdot \mathbf{u} = 0 \tag{10}$$

Momentum:

$$\frac{D\mathbf{u}}{Dt} = \frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} \tag{11}$$

or, in terms of vorticity

$$\frac{D\xi}{Dt} = \nu\nabla^2\xi \tag{12}$$

where, by definition $\xi \mathbf{e}_x = \nabla \times \mathbf{u}$.

If viscosity is neglected, Eq. (12) reduces to

$$\frac{D\xi}{Dt} = 0 \tag{13}$$

The stream function, Ψ , is defined through the equality

$$\mathbf{u} = \nabla \times \Psi \mathbf{e}_x \tag{14}$$

Combining Eqs. (14) and (10) with the definition of $\xi \mathbf{e}_x$, the continuity equation in terms of the stream function becomes

$$\nabla^2 \Psi = -\xi \tag{15}$$

The boundary condition and the far field condition for an inviscid flow are

$$\mathbf{u} \cdot \mathbf{n} = 0 \qquad \text{on wing surface} \tag{16a}$$

$$\mathbf{u} \rightarrow \mathbf{U}_\infty \qquad \text{as } \mathbf{x} \rightarrow \infty \tag{16b}$$

In terms of stream function

$$\Psi = \text{constant} \qquad \text{on wing surface} \tag{17a}$$

$$\Psi = \Psi_\infty \qquad \text{as } \mathbf{x} \rightarrow \infty \tag{17b}$$

The governing equations are replaced by Eq. (15) and Eq. (13) with conditions given by Eqs. (17a,b).

2.3 Vortex Model

In this study the rolled-up shear layer is replaced by a cloud of discrete vortices having finite core size, each of which is generated from the separation point at a time step. Chorin[27] first introduced the concept of vortices in the cloud approach having finite core size, so that all of the rotational and viscous effects can be assumed to take place inside the core, while outside of this core the flow field is governed by conventional potential theory. Using

this concept, the vorticity distribution, which appears in the right hand side of Eq. (16), can be represented as:

$$\xi(\mathbf{x},t) \;=\; \sum_{i=1}^{N} \Gamma_i \; \gamma_i(|\mathbf{x}-\mathbf{x}_i(t)|) \tag{18}$$

where γ_i is the vorticity distribution function satisfying the normalizing condition

$$\int_0^\infty \gamma_i(\mathbf{x})\, d\mathbf{x} \;=\; 1 \tag{19}$$

In case of point vortices, the functions γ_i's become the dirac-delta function $\delta(\mathbf{x}-\mathbf{x}_i)$. Introducing a core radius σ_i, Eq. (18) can be written as:

$$\gamma_i(|\mathbf{x}-\mathbf{x}_i|) \;=\; \frac{1}{{\sigma_i}^2}\, f\Big(\frac{|\mathbf{x}-\mathbf{x}_i|}{\sigma_i}\Big) \tag{20}$$

where f is a shape function common to all vortices.

The velocity induced by the vorticity distribution $\Gamma_i\gamma_i(|\mathbf{x}-\mathbf{x}_i(t)|)$ is

$$\mathbf{u}(\mathbf{x},t) \;=\; -\frac{1}{2\pi}\frac{(\mathbf{x}-\mathbf{x}_i)\times \mathbf{e}_x\, \Gamma_i\, g\Big(\frac{|\mathbf{x}-\mathbf{x}_i|}{\sigma_i}\Big)}{|\mathbf{x}-\mathbf{x}_i|^2} \tag{21}$$

where

$$g(y) \;=\; 2\pi \int_0^y f(z)\; z\; dz \tag{22}$$

The following distributions, first used by Spalart,[33] will be applied here

$$\overline{U} \;=\; \frac{2\pi\, U\, \sigma}{\Gamma} \;=\; \left\{ \begin{array}{cc} \dfrac{1}{\overline{r}} & \overline{r} \;\geq 1 \\ \overline{r}\,(3 \;-\; 3\,\overline{r}^2 \;+\; \overline{r}^4) & \overline{r} \;\leq\; 1 \end{array} \right\} \tag{23}$$

$$\pi\, f \;=\; \left\{ \begin{array}{cc} 0 & \overline{r} \;\geq\; 1 \\ 3(\overline{r}^2 \;-\; 1)^2 & \overline{r} \;\leq\; 1 \end{array} \right\} \tag{24}$$

where $\bar{r} = r/\sigma$.

These distributions are shown in Fig. 4. The vorticity distribution is directly related to the vorticity diffusion. Prandtl[44] studied vorticity diffusion by solving the diffusion equation and found that

$$\sigma^2 \propto t \tag{25}$$

valid for a Gaussian distribution of vorticity. Here, distributions given by Eqs. (23) and (24) are used together with the diffusion relationship given by Eq. (25).

2.4 Kutta Condition

In this approach, a new discrete vortex is generated from the separation point at each time step such that the flow field including this new vortex satisfies an appropriate Kutta condition at the separation point. For an unsteady flow, as opposed to a steady flow, the vorticity is supplied to the surrounding fluid continuously from a separation point such that the strength and the position of the shed vorticity allow for pressure continuity to be satisfied. Such strength and position are functions of the velocity on each side of the vortex sheet. From Fig. 5a, Bernoulli's equation on each side of the vortex sheet is

$$H_u = \frac{1}{2}\rho\, {u_u}^2 + p_u + \rho \frac{\partial \phi_u}{\partial t} \tag{26a}$$

$$H_l = \frac{1}{2}\rho\, {u_2}^2 + p_l + \rho \frac{\partial \phi_l}{\partial t} \tag{26b}$$

Here, H, u and p represent the total head, velocity and pressure respectively. The total head must be the same on each side since the whole flow is started from a uniform state. This requirement, together with pressure continuity, gives

$$\left|\frac{\partial(\phi_l - \phi_u)}{\partial t}\right| = \frac{1}{2}|({u_l}^2 - {u_u}^2)| \tag{27}$$

Since the difference of velocity potential across the sheet represents the vortex intensity of the sheet, there results

$$\left|\frac{\partial \gamma}{\partial t}\right| = \frac{1}{2}\left|({u_l}^2 - {u_u}^2)\right| \tag{28}$$

The separation angle of the vortex sheet, θ_s, is such that the boundary condition on the wing surface, given by Eq. (17a), is satisfied. In this study the separation angle is set to zero, as shown in Fig. 5b, implying that vorticity going into the vortex structure is shed parallel to the lower surface.[40] This assumption makes the velocity on the upper side of the vortex sheet equal to zero, since the assumed streamline, as indicated in Fig. 5b,causes a stagnation point on the upper part of the vortex sheet. Combining Eq. (28) with the transformation $x = U \cos \alpha \, t$, the strength of a new vortex can be calculated.

$$\Delta\gamma = \Gamma_{\text{new}} = \frac{1}{2} {u_l}^2 \, \Delta t \tag{29a}$$

$$= \frac{1}{2} {u_l}^2 \frac{\Delta x}{U \cos \alpha} \tag{29b}$$

The position of this new vortex can be obtained by considering the convection velocity at the separation point. By definition, a vortex sheet moves with the average of the velocities at its two sides. Then, the convection velocity of the newly shed vortex is

$$U_s = \frac{1}{2}(u_l + u_u) = \frac{1}{2} u_u \tag{30}$$

The position of the new vortex is then

$$z_n = z_s + \frac{\mathbf{S}_s}{|\mathbf{S}_s|} U_s \, \Delta t = z_s + \frac{\mathbf{S}_s}{|\mathbf{S}_s|} U_s \frac{\Delta x}{U \cos \alpha} \tag{31}$$

where z_s is the position of the separation point, and $\mathbf{S}_s$ is the slope at the separation point, as illustrated in Fig. 5c.

2.5 Wing Cross-Section Representation

The problem consists in solving Eq. (16) with the vorticity distribution given by Eq. (18) on the rolled-up shear layer, subject to boundary conditions (17a,b) and the Kutta condition expressed by Eqs. (29) and (31). In this study the wing cross-section is represented by vorticity panels with linearly varying strength. The body is divided into m panels and at n^{th} time step there are n vortices in the wake, as shown in Fig. 6. The stream function at the i^{th} panel can be written as

$$\Psi_i = \int \gamma(s_j)\, K(s_i, s_j)\, ds_j + \frac{1}{2\pi}\sum_{k=1}^{n}\Gamma_k \ln r_{ik} + U \sin\alpha\, y_i \tag{32}$$

with $\gamma(s_j)$ the bound vortex strength at the j^{th} panel, and $K(s_i, s_j)$ the geometric influence coefficient, which depends only wing geometry. The second term on the right hand side represents the stream function due to the discrete vortices distributed on the shear layer. r_{ik} is the distance between the i^{th} panel and the k^{th} vortex. The last term in Eq. (22) is due to the free stream. In this approach the relevant velocity is the vertical component in the cross-flow plane. Boundary condition (17a) gives

$$\Psi_1 = \Psi_2 = \cdots = \Psi_m = \text{const.} \tag{33}$$

Conservation of vorticity implies that the sum of the vorticity in the whole flow field must be zero

$$\int \gamma_j\, ds_j + \sum_{k=1}^{n}\Gamma_k = 0 \tag{34}$$

Combining Eqs. (32), (33) and (34), the following system of equations results

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} & 1 \\ a_{21} & a_{22} & \cdots & a_{2m} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} & 1 \\ s_1 & s_2 & \cdots & s_m & 0 \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_m \\ c \end{pmatrix} = - \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \\ \sum_{k=1}^{n}\Gamma_k \end{pmatrix} \tag{35}$$

The matrix a_{ij} is called the geometric influence coefficient matrix and is constant during time iteration. The last row in the matrix represents the integration of the bound vortex strength. Using a trapezoidal rule for the integration, s_j can be written as

$$s_j = \frac{1}{2}(l_j + l_{j+1}) \tag{36}$$

where l_j is the length of j^{th} panel. c is the constant value of the stream function representing the wing surface, and the b_j's are the stream function values induced by the wake and the free stream. The computational implementation of matrix a_{ij} is described in Lee's work[45] Eq. (35) is solved until the solution (the vortex position and strength and the bound vortex strength) reach a steady state. Only the left-hand-side vector is varied at each time step by increasing the number of vortices in the shear layer, while the right-hand-side matrix is kept constant.

2.6 Wing Load Distribution

The pressure distribution on wing surfaces for a 2-D unsteady flow field can be obtained using the unsteady Bernoulli's equation

$$p_\infty + \frac{1}{2}\rho U^2 \sin^2\alpha = p + \frac{1}{2}\rho v_c^2 + \rho \frac{\partial \phi}{\partial t} \tag{37}$$

where v_c is the velocity on the wing surface

Using the transformation $x = U \cos\alpha\, t$, the last term in Eq. (37) can be rewritten as

$$\frac{\partial \phi}{\partial t} = U \cos\alpha \frac{\partial \phi}{\partial x} \tag{38}$$

Then the pressure coefficient is

$$\begin{aligned} C_p &= \frac{p - p_\infty}{\frac{1}{2}\rho U^2} \\ &= \sin^2\alpha - \left(\frac{v_c}{U}\right)^2 - 2\cos\alpha \frac{1}{U}\frac{\partial \phi}{\partial x} \end{aligned} \tag{39}$$

This is equivalent to the first order term of the pressure coefficient obtained from slender body theory, which gives

$$\begin{aligned} C_p &= 1 - \left(\frac{v_c}{U}\right)^2 - \left(\cos\alpha + \frac{u}{U}\right)^2 \\ &= 1 - \left(\frac{v_c}{U}\right)^2 - \cos^2\alpha - 2\frac{u}{U} - \left(\frac{u}{U}\right)^2 \\ &\sim \sin^2\alpha - \left(\frac{v_c}{U}\right)^2 - 2\frac{u}{U}\cos\alpha \end{aligned} \tag{40}$$

The velocity component v_c is numerically equal to the bound vortex strength per unit length, since the flow internal to the wing cross-section is assumed to be at rest. As indicated in Fig. 7, the bound vortex strength can be written as

$$\gamma\, ds = \int \mathbf{v} \cdot d\mathbf{l} = v_c\, ds \tag{41}$$

hence

$$\gamma = v_c \tag{42}$$

In this study, the bound vortex strength itself represents the outer velocity of the boundary layer. As a result, the velocity v_c in the cross-flow plane can be obtained from the solutions of Eq. (35). The perturbation velocity in the axial direction can be obtained from the following conical relationship

$$u = \frac{1}{x}\left(\Phi - y\frac{\partial\Phi}{\partial y} - z\frac{\partial\Phi}{\partial z}\right) \tag{43}$$

The velocity potential is given in terms of logarithmic functions and its evaluation requires consideration of branch cuts, as shown in Fig. 8.

By integrating the load distribution given by Eq. (39) over wing and flap surfaces, the total resultant force on each surface can be obtained

$$R_{\mathrm{W,\,F}} = \int_{\mathrm{W,\,F}} p\, ds \tag{44}$$

where W and F represent wing and flap surfaces respectively.

These resultant forces can be decomposed into lift and drag

$$L_W = R_W \cos\alpha \tag{45a}$$

$$D_W = R_W \sin\alpha \tag{45b}$$

$$L_F = R_F (\cos\delta + \tan\epsilon \sin\delta \sin\alpha) \tag{45c}$$

$$D_F = R_F (\cos\delta \sin\alpha - \tan\epsilon \sin\delta \cos\alpha) \tag{45d}$$

The lift-to-drag ratio of the wing is then

$$\frac{L}{D} = \frac{1}{\alpha}\left[\left(1 + \frac{R_{Fy}}{R_w + R_{Fy}} \frac{\epsilon}{\alpha} \tan\delta\right) + O(\alpha^2)\right] \tag{46}$$

where R_w and R_{Fy} represent the force components acting on wing and flap surfaces, as shown in Fig. 9. The second term inside the bracket shows the increment in L/D due to flap deflection, which is of the same order of magnitude as the lift-to-drag ratio without flap deflection. Detailed derivations are given in Ref (23).

2.7 Limits of Flap Deflection Angle

The present study uses a cross-flow plane analysis, which implies that the significant velocity component causing separation along leading edges is the component normal to the wing surface. The maximum flap deflection compatible with this approach is related to the velocity component normal to the flap surface. If that component is directed towards the upper flap the surface, as shown in Fig. 10, the vorticity system tends to be located under the wing, as demonstrated by experiments. Experimental observations on a model of aspect ratio 2.31[21] showed that the vortical system was located under the wing surface when the angle of attack was larger than the flap deflection angle. In general, the ratio α/ϵ for which this situation occurs is expected to depend on the wing aspect ratio. The cross-flow plane analysis cannot, by definition, exactly account for the actual orientation

of the velocity component normal to the flap surface. Since the discrepancy between the actual velocity component normal to the flap surface and the one assumed by the cross-flow analysis increases with flap deflection, this method is expected to fail for flap deflections beyond a certain range. An assessment of that range can be made by considering under which conditions the velocity component normal to the flap surface vanishes (a situation unaccountable for in the cross-flow analysis.) This assessment is facilitated if the velocity component normal to the flap surface is assumed to be due to the free stream velocity alone, that is, if the perturbation velocity of the wing is neglected. This is plausible if the angle of attack and apex angles are both small.

The outward normal vector, $\mathbf{n}$, on the flap surfaces is

$$\mathbf{n} = -\sin\delta \sin\epsilon \, \mathbf{e}_x + \cos\delta \, \mathbf{e}_y + \cos\epsilon \sin\delta \, \mathbf{e}_z \tag{47}$$

The free stream velocity has two components such that

$$\mathbf{U}_\infty = U \cos\alpha \, \mathbf{e}_x + U \sin\alpha \, \mathbf{e}_y \tag{48}$$

The component of the free stream normal to the flap surface is then

$$U_n = \mathbf{U}_\infty \cdot \mathbf{n} = -U \sin\delta \sin\epsilon \cos\delta + U \sin\alpha \cos\delta \tag{49}$$

The condition that $U_n > 0$ is

$$\frac{\tan\alpha}{\sin\epsilon} > \tan\delta \tag{50}$$

The boundary shown in Fig. 11 represents the limit where the component of the free stream velocity normal to the flap surface vanishes for $\epsilon = 22$ deg..

2.8 Trailing Edge Wake

The trailing edge wake after a delta wing, as shown in Fig. 1, has a shape different from that of a conventional large aspect-ratio wing. The pressure peak just below the leading edge vortex core causes a vorticity distribution such that the bound vortex strength, as a function of spanwise location, has a maximum between the wing center and the wing leading edge, as illustrated in Fig. 12. In this study the conical flow is treated under the assumption that the wing extends infinitely far downstream, namely, no trailing edge exists. To study the trailing edge wake roll-up in a qualitative way, an artificial trailing edge is assumed to exist, from which the conical vorticity distribution calculated on the wing surface is released . A Kutta condition is applied such that there is no pressure jump at the trailing edge.

The bound vorticity on the wing surface has two components, as shown in Fig. 13. These two components must satisfy the divergence free condition

$$\nabla \cdot \gamma = \frac{\partial \zeta}{\partial x} + \frac{\partial \eta}{\partial z} = 0 \tag{51}$$

where $\gamma = \zeta \, \mathbf{e}_x + \eta \, \mathbf{e}_z$

The absence of a pressure jump at the wing trailing edge demands that the shed vorticity at the trailing edge should be aligned with the local velocity. An approximation form of the trailing edge Kutta condition can be written as follows

$$\frac{\partial \eta}{\partial z} = 0 \tag{52}$$

then Eq. (51) reduces to

$$\frac{\partial \zeta}{\partial x} = 0 \tag{53a}$$

or

$$\zeta = \zeta(z) \tag{53b}$$

This indicates that ζ , which is identified with the γ's from Eq. (35), is constant as it moves downstream. After the trailing edge, the evolution of the wake is solved in a straightforward way using a 2-D time evolution technique.

3. NUMERICAL IMPLEMENTATION

3.1 Kutta Condition

The flow separation in a 2-D or a 3-D flow arises from viscous effects. An equivalent effect is imposed on the inviscid problem through a Kutta condition. A converged quasi-steady analysis requires that the shed vorticity at the wing side edges be convected with finite velocity. This velocity, not known a priori, is the average of the velocity on both sides of the wing at the side edge.[38] In a 2-D case the velocity at the separation point can be determined from the experimentally established fact that the Strauhal number, when expressed in terms of the base pressure at the separation point, is constant.[30] Experimental observations on the flow past an inclined flat plate have shown that this velocity is about 1.5 U_∞ .[30] Under the conical flow assumption, the value of this velocity must be constant along the leading edges. Other workers have attempted to find this value experimentally and applied it to the solution of delta wing problems with the vortex cloud method,[12] failing to attain converged solutions. In this work this velocity is determined by recognizing the fact that, after a sufficiently large time, it must achieve a constant value, for given α, ϵ and δ . It was found that if an initial value for the bound vortex strength of about $3U_\infty$ is assumed, convergence to the final value of the side edge convective velocity is achieved within 300 steps. Too large initial values lead to oscillatory end behavior, while to small initial values require significantly more time steps.

3.2 Core Model

The core region for a conical flow has large axial velocity and the flux of vorticity generated along the leading edges and supplied to the core through the rolled-up shear layer is balanced by that convected by the axial velocity. Thus, the core region, usually identified with the "vortex jungle" for a 2-D vortex cloud method, cannot be included in this model

of the conical flow since it would violate the condition of aerodynamic slenderness. Hence, the core region is represented by a single vortex through a simple merging scheme. By imposing a restriction on the rotation angle of the vortices, the efficiency of the numerical effort is increased and good definition of shear layer is achieved. Other studies[8,16] have shown that the rotation angle of the shear layer, as long as it is set greater than 2π, has a small effect on overall quantities. The same conclusion is arrived at in this study.

The vortices which rotate more than a given angle are allowed to merge to the core, which is then simulated by a single vortex. The position, strength and core size are determined such that the angular momentum before and after the merging is conserved. If k vortices are merged together, conservation of angular momentum outside the core region is expressed by the following relationships for the core position and strength

$$z_c = \frac{\sum_{i=1}^{k} z_i \, \Gamma_i}{\sum_{i=1}^{k} \Gamma_i} \tag{54}$$

$$\Gamma_c = \sum_{i=1}^{k} \Gamma_i \tag{55}$$

The angular momentum inside the core is

$$M = \int_0^{2\pi} \int_0^{\sigma} (\mathbf{r} + \rho) \, \mathbf{U} \, \rho \, d\theta \, d\rho \tag{56}$$

where $\mathbf{r}$ is the position vector from the origin to the core center, and ρ is a vector from the core center to a point inside the core. U is the velocity distribution inside the core as given by Eq. (23). Performing the integration, the angular momentum reduces to

$$M = \frac{17}{24} \Gamma \, \sigma^3 \tag{57}$$

Then the core size after merging is

$$\sigma_c{}^3 = \frac{\sum_{i=1}^{k} \Gamma_i \, \sigma_i{}^3}{\sum_{i=1}^{k} \Gamma_i} \tag{58}$$

The strength and position are independent of the vorticity distribution inside the core, but the core size is dependent on the vorticity distribution.

3.3 Merging between Neighboring Vortices

A similar merging technique as the one used in the core model is also applied to the neighboring vortices such that if the distance between them is less than a given value two vortices are allowed to merge into one. This value is set equal to the mean value of the radii of two vortices. The position, strength and the core radius after merging are determined from Eqs. (54), (55) and (58). The merging scheme allows for a reduction of the total number of vortices by almost a half, making the scheme much more efficient.

3.4 Merging of Vortices on Wing Surfaces

In this discretization of a separated wake, vortices coming very close to the wing surface have the tendency to penetrate the wing surface. This is observed to happen even with very small time steps and large number of panels. A means of fixing this difficulty has been suggested by Lighthill[46], and consists in allowing the vortices that come within a thin layer on the wing surface to disappear. Another way of solving this difficulty is by relocating a vortex that penetrated the surface in a preceding time step on its image point outside the surface.[35] In this study the vortices which come closer to the wing surface than a given distance, which is taken as 0.5% of the half span length, are made to disappear. The effect of the merged vortex appears through Eq. (34) by rearranging the bound vortex

distribution for the next time step. This surface merging occurs when the angle of attack is small or when the flap deflection angle is large.

3.5 Integration Method For Advancing Vortex Position

Any 2-D discrete vortex method contains a vorticity diffusion effect due to the numerical error involved in time integration. The numerical diffusion effect is shown in Fig. 14. Some authors[29,31,32] maintain that this numerical error could be responsible for instability of the vortex motion. Others[32] suggest that this numerical error simulates a turbulence diffusion rather than a viscous diffusion. In a conical flow the diffusion effect arising from the numerical error is less marked than in a 2-D flow. This may be attributed to the fact that the shear layers in the conical flow develop in a field of intense circumferential velocity, which allows for a shorter time during which numerical dissipation acts. Here both a 1st order Euler method and a 4th order Runge-Kutta method were used for time integration. These schemes are:

1st order Euler method

$$Z_v(t+\Delta t) \;=\; Z_V(t) \;+\; \mathbf{U}_v(t, Z_v(t))\Delta t \tag{59}$$

4th order Runge-Kutta method

$$\begin{aligned}
Z_v^*(t+\Delta t/2) &= Z_v(t) + \frac{\Delta t}{2}\,\mathbf{U}_v(t, Z_v(t)) \\
Z_v^{**}(t+\Delta t/2) &= Z_v(t) + \frac{\Delta t}{2}\,\mathbf{U}_v(t+\Delta t/2, Z_v^*(t+\Delta t/2)) \\
Z_v^{***}(t+\Delta t) &= Z_v(t) + \Delta t\;\mathbf{U}_v(t+\Delta t/2, Z_v^{**}(t+\Delta t/2)) \\
Z_v(t+\Delta t) &= Z_v(t) + \frac{\Delta t}{6}\Big(\mathbf{U}_v(t, Z_v(t)) + 2\,\mathbf{U}_v(t+\Delta t/2, Z_v^*(t+\Delta t/2)) \\
&\qquad + 2\,\mathbf{U}_v(t+\Delta t/2, Z_v^{**}(t+\Delta t/2)) + \mathbf{U}_v(t+\Delta t, Z_v^{***}(t+\Delta t))\Big)
\end{aligned} \tag{60}$$

4. RESULTS

According to the argument given in chapter 2, the range of angle of attack was chosen of same order as the wing apex angle. For comparison with experiments three values of angle of attack were selected, α = 15, 25, 35 deg. This satisfies the requirement $O(\alpha) = O(\epsilon)$ for the experimental model used in Ref. (43).

Fig. 15 shows the convergence of the vortex strength at the separation point for an initial guess of vortex strength of $3U_\infty$. Significant oscillation can be seen during the first 50 steps. If too large an initial guess is used, the vortex strength oscillates such that it changes sign during some time steps, leading to nonphysical results. A proper initial guess for each case, given α and δ, is needed to obtain the final vortex strength at the separation point.

Fig. 16a shows the vortex jungle that arises when no core model is used. In addition, the shear layer is not clearly defined. Fig. 16b shows a case where a core rotation angle of 2.5π was imposed for the same configuration and shows a good definition of the shear layer.

Fig. 17 shows the comparison between Smith's calculation[8] and this theory for the core position and the geometry of the rolled-up shear layer, for $\alpha/\epsilon = 1$.

A comparison between results from a 3-D panel code analysis[24] and the present method is illustrated in Fig. 18 for the case of a leading edge fence, corresponding to an upward deflection of the leading edge flap of more than 90 deg. The results form the 3-D panel analysis were chosen at the 30% chord from the apex to minimize the upstream effect of the trailing edge. Fig. 18a is the case without the fence. The pressure peak point calculated by this analysis is closer to the leading edge as compared with the panel results. The case of 130 deg. fence angle is shown in Figs. 18b,c. The results show better agreements between both calculations for the case of deflected fence. The present study reveals a narrower

pressure peak region.

The data for the comparison with experiments were taken from Ref. (43), where the model had a thickness at the trailing edge of 7.7% of the half span and a 22 deg. half apex angle for undeflected flap. The ratio of the total span to that of the main wing was 0.61. The data used for comparison were measured at a spanwise station located 37% of the chord from the apex, again to minimize the effect of trailing edge..

Figs. 19a,b show the rolled-up shear layer and surface pressure distribution for α =25 deg. and $\delta = 0$ deg. Agreement can be considered very good, except for a small region near the leading edge, where a secondary vortex may have been present in the experiments and would be responsible for the disagreement.

Figs. 20 and 21 are the cases for flap deflection angle of 15 deg. and 30 deg. and same angle of attack as the case of Fig. 19. The pressure at the centerline decreases with flap angle, which indicates that the effect of flap deflection is similar to a reduction of angle of attack. Although the vortex position does not move much with flap angle, some movement toward the surface is observed, which has also been reported elsewhere.[23] The hinge vortex is hardly observable in the cases analyzed here, being quite weak and tending to be absorbed on the wing surface. Fig. 21 shows a weak hinge vortex. The greater discrepancy between computation and experiment near the leading edge is attributed to the stronger secondary vortices at higher flap deflections.

Fig. 22 shows the case of $\alpha = 15$ deg., without flap deflection. The pressure distribution shows good agreement. The effect of a secondary vortex is also stronger than that at α =25 deg.

Fig. 23 shows the rolled-up shear layer and surface pressure distribution when α =35 deg., without flap deflection. The shear layer shows a tendency to produce a type of coherent structures which have been experimentally observed, as described in Fig. 2, taken from Ref. (25)

The convergence of the present calculation can be assessed through core position, core

vortex strength and core radius, as shown in Figs. 24a,b. The core vortex strength shows some oscillation even after convergence is attained, since the vortices in the shear layer merge into the core with some time interval.

Fig. 25 shows the effect of two schemes of time integration. As anticipated in chapter 3, the 1^{th} order Euler scheme gives a much more diffused shear layer than the 4^{th} order Runge-Kutta method. Overall flow field features do not depend significantly on the integration scheme.

The effect of the merging scheme between neighboring vortices is shown in Figs. 26a,b. By replacing high density clouds with fewer vortices, the total number of vortices is reduced by almost a half.

The calculated trailing edge wake is compared with computations using a higher-order panel method[47] for three different time steps. As can be seen in Fig. 27, good agreement is obtained.

Fig. 28 shows the effect of core rotation angle on the pressure distribution for the case of the leading edge fence of Fig. 18. Four core rotation angles were studied. The vortex strength decreases with increasing core rotation angle, with values greater than π not exhibiting any significant change in vortex strength for this configuration. Even large variations of rotation angle cause little change in pressure distribution.

5. CONCLUSIONS AND RECOMMENDATIONS

A conical delta wing with leading edge flaps was analyzed using a modified 2-D vortex cloud method based on a mathematical analogy between 3-D conical flows and 2-D unsteady self-similar flows. To avoid the problem of high vortex stretching due to large axial velocities in the core region, a simple core model was used. Previous work has shown that the effect of flap deflection is similar to a decrement of angle of attack. This has been confirmed in this work. The distortion of the rolled-up shear layer as a result of flap deflection was also studied. With simple and shear layer core models good shear layer definition and satisfactory agreement between theory and experiments in surface pressure distribution was achieved.

For the range of angle of attack analyzed here there is a negligible hinge vortex. The experimentally observed instabilities of the shear layer is hardly found, except in the high angle of attack range, and becomes more visible if no merging scheme between neighboring vortices is implemented. Use of a 1^{st} order Euler scheme for time integration shows more clearly this kind of instability than a 4^{th} order Runge-Kutta method.

The effect of secondary vortices becomes significant in the range of small angle of attack and large flap deflection angle. The present study cannot describe this phenomenon which is due to viscous effects. A possible way of accounting for this problem is to interact between the inviscid problem and boundary layer model which can predict 3-D boundary layer separation. Such viscous/inviscid approach would constitute an interesting extension of this work.

A maximum value of flap deflection exists, beyond which this analysis becomes invalid. Beyond this limit full 3-D solutions of the equations of motion would be required.

The present study gives a good definition of shear layers, which current Navier-Stoke's solvers cannot provide since an impractically fine grid distribution would be required to

capture such discontinuity surfaces. This work can provide some insight in supplying initial grid distributions to Navier-Stoke's solvers, thereby reducing computational cost.

A recent study on the structures of rolled-up shear layers[48] has shown that the vortical system could be stabilized by altering the spacing of coherent structure in shear layers. Such a concept for stabilizing the rolled-up shear layers could be further explored with the methodology developed here.

REFERENCES

1. R.T. Jones, "Properties of Low-Aspect-Ratio Pointed Wings at Speeds Below and Above the Speed of Sound, "NACA Rep. No. 835, 1946
2. R. Legendre, "The Kutta-Joukowsky Condition in Three-Dimensional Flow, "RAE LT 1709, 1973
3. C.E. Brown and W.H. Michael, Jr., "On Slender Delta Wings with Leading-Edge Separation, "NACA TN 3430, 1955
4. K.W. Mangler and J.H.B. Smith, "Theory of Slender Delta Wings with Leading-Edge Separation, "Proc. Roy. Soc. London Ser. A, Vol. 251, pp. 200-, 1959
5. D. Küchmann and J. Webber, "Vortex Motion, "Zeithschrift Für Angewandte Mathmatik und Mechanik, 45, pp. 457-474, 1965
6. H. Ashley and M. Landahl, Aerodynamics of Wings and Bodies, Addison Wesely Publishing Co. 1965
7. E.G. Polhamus, "A concept of the Vortex Lift of Sharp Edge Delta Wings Based on a Leading-Edge Suction Analogy, "NASA TN-D-3767, 1966
8. J.H.B. Smith, "Improved Calculation of Leading-Edge Separation Form Slender, Thin, Delta Wings, "Proc. Roy. Soc. Ser. A, Vol. 306, 1968
9. J.E. Barsby, "Flow Past Conically Cambered Slender Delta Wings with Leading-Edge Separation, "ARC R&M No. 3748, 1972
10. J.H.B. Smith, "The Isolated-Vortex Model of Leading-Edge Separation Revised for Small Incidence, "RAE TR 73160, 1973
11. J.H.B. Smith, "Theoretical Modelling of Three Dimensional Vortex Flow in Aerodynamics, "Aeronautical Journal, Jan. 1973
12. M.R. Mendenhall and J.N. Nielsen, "Effect of Symmetrical Vortex Shedding on the Longitudinal Aerodynamic Characteristics, "NASA CR-2473, 1975

13. D. Küchmann, The Aerodynamic Design of Aircraft , Pergamon Press, 1979

14. D. Hummel, "On the Vortex Formation over a Slender Wing at Large Angle of Incidence, "AGARD CP 147

15. J.H.B. Smith, "Inviscid Fluid Model, Based on Rolled-up Vortex Sheets, for Three-Dimensional Separation at High Reynolds Number, "AGARD LS 94, 1978

16. A.J. Peace, "A Multi-Vortex Model of Leading-Edge Vortex Flows, "International Journal for Numerical Methods in Fluids, Vol. 3, 1983

17. P.L. Coe, Jr. and R.P. Weston, "Effects of Wing Leading-Edge Deflection on Low-Speed Aerodynamics Characteristics of a Low-Aspect-Ratio Highly Swept Arrow-Wing Configuration, "NASA TP 1434, 1979

18. J.F. Marchmann, III, "The Aerodynamics of Inverted Leading Edge Flaps on Delta Wings, "AIAA Paper 81-0356, 1981

19. D.M. Rao and T.D. Johnson, Jr., "Subsonic Pitch-Up Alleviation on a 74 deg. Delta Wing, "NASA CR-165749, 1981

20. L.P. Yip and D.G. Murri, "Effect of Vortex Flaps on the Low-Speed Aerodynamics of an Arrow Wing, "NASA TP 1914, 1981

21. D.M. Rao, "Upper Vortex Flap - A Versatile Surface for Highly Swept Wings,"ICAS-82-6.7.1, 1982

22. J.E. Lamar and J.F. Campbell, "Vortex Flaps - Advanced Control Devices for Supercruise Fighters, "Aerospace America, Jan. 1984

23. S. Oh and D. Tavella, "Analytical Observations on the Aerodynamics of a Delta Wing with Leading Edge Flaps, "JIAA TR-74, 1986

24. C.S. Reddy, "Spanwise Pressure Distribution on Delta Wing with Leading-Edge Vortex Flap, "J. of Aircraft, Vol. 24, No. 3, pp. 222-234, 1987

25. F.M. Payne, T.T. Ng and R.C. Nelson, "Visualization and Flow Surveys of the Leading Edge Vortex Structure on Delta Wing Planforms, "AIAA Paper 86-0330, 1986

26. L. Rosenhead, "The Formation of Vortices from a Surface Discontiuity, "Proc. Roy. Soc. London Ser. A, Vol. 134, pp. 170-192, 1931

27. A.J. Chorin, "Numerical Study of Slightly Viscous Flow, "JFM Vol. 57, pp. 785-796, 1973

28. T. Sarpkaya, "An Inviscid Model of Two-Dimensional Vortex Shedding for Transient and Asymptotically Steady Separated Flow over an Inclined Plate, "JFM, Vol. 68, pp. 109-128, 1975

29. R.R. Clements and D.J. Maull, "The Representation of Sheets of Vorticity by Discrete Vortices, "Prog. Aerospace Sci. Vol. 16, pp. 129-146, 1975

30. A. Fage and F.C. Johansen, "On the Flow of Air Behind an Inclined Flat Plate of Infinite Span, "Proc. Roy. Soc. Ser. A, Vol. 363, 1978

31. P.T. Fink and W.K. Soh, "A New Approach to Rolled-Up Calculation of Vortex Sheets, "Proc. Roy. Soc. London Ser. A, Vlo. 362, 1978

32. K. Kuwahara, "Study of Flow Past a Cylinder by an Inviscid Model, "Journal of the Physical Society of Japan, Vol. 45, No. 1, 1978

33. P.R. Spalart and A. Leonard, "Computation of Separation Flows by a Vortex-Tracing Algorithm, "AIAA Paper 81-1246, 1981

34. P.M. Stremel, "A Method for Modelling Finite Core Vorticies in Wake Flow Calculation, "AIAA Paper 84-0417, 1984

35. T. Fujinami, G.S. Dulikravich and A.A. Hassan, "Free-Vortex Method Simulation of Unsteady Airfoil/Vortex Interaction, "AIAA Paper 86-1792, 1986

36. D.J. Lee and C.A. Smith, "Distorsion of the Vortex Core During Blade/Vortex Interaction, "AIAA Paper 87-1243, 1987

37. B.C. Basu and G.J. Hancock, "The Unsteady Motion of a Two-Dimensional Aerofoil in Incompressible Inviscid Flow, "JFM Vol. 87, pp. 159-178, 1978

38. R.I. Lewis, "Surface Vorticity Modelling of Separated Flows From Two-Dimensional Bluff Bodies of Arbitrary Shape, "Journal of Mechanical Engineering Science, Vol.

23, No.1, pp. 1-12, 1981

39. P.K. Stansby, "A Generalized Discrete-Vortex Method for Sharp-Edged Cylinder, "AIAA Journal, Vol 23, No.6, pp. 856-860, 1985
40. D.R. Polling and D.P. Telionis, "The Response of Airfoils to Periodic Disturbances - The Unsteady Kutta Condition, "AIAA Journal, Vol. 24, No. 2, pp. 193-199, 1986
41. F.T. Johnson, E.N. Tinoco, P. Lu and M.A. Epton, "Three-Dimensional Flow over Wings with Leading-Edge Vortex Separation, "AIAA Journal, Vol. 18, No. 4, pp. 367-380, 1980
42. O.A. Kandil, "Recent Improvements in the Prediction of the Leading and Trailing Edge Vortex Cores of Delta Wings, "AIAA Paper 81-1263, 1981
43. C.S. Lee "Experimental Studies of a Delta Wing with Leading Edge Flaps, "JIAA TR-76, 1986
44. L. Prandtl, "The Mechanics of Viscous Flow "in Aerodynamic Theory Vol. 2, W.F. Durand. Berlin pp. 61-69, 1935
45. C.S. Lee and S. Bodapati, "Calculation of the Unsteady Flow Field of an Airfoil with a Deflected Spoiler by Vortex Method, "JIAA TR-62, 1985
46. M.J. Lighthill, Chapter II in L. Rosenhead Laminar Boundary Layers, Clarendon Press, 1963
47. H.W.M. Hoeijmakers and W. Vaatstra, "A Higher Order Panel Method Applied to Vortex Sheet Roll-Up, "AIAA Journal, Vol. 21, No. 4, pp. 516-523, 1983
48. M. Gad-el-Hak and R.F. Blackwelder, "Control of the Discrete Vortices form a Delta Wing ", AIAA Journal, Vol. 25, No. 8, pp. 1042-1049, 1987

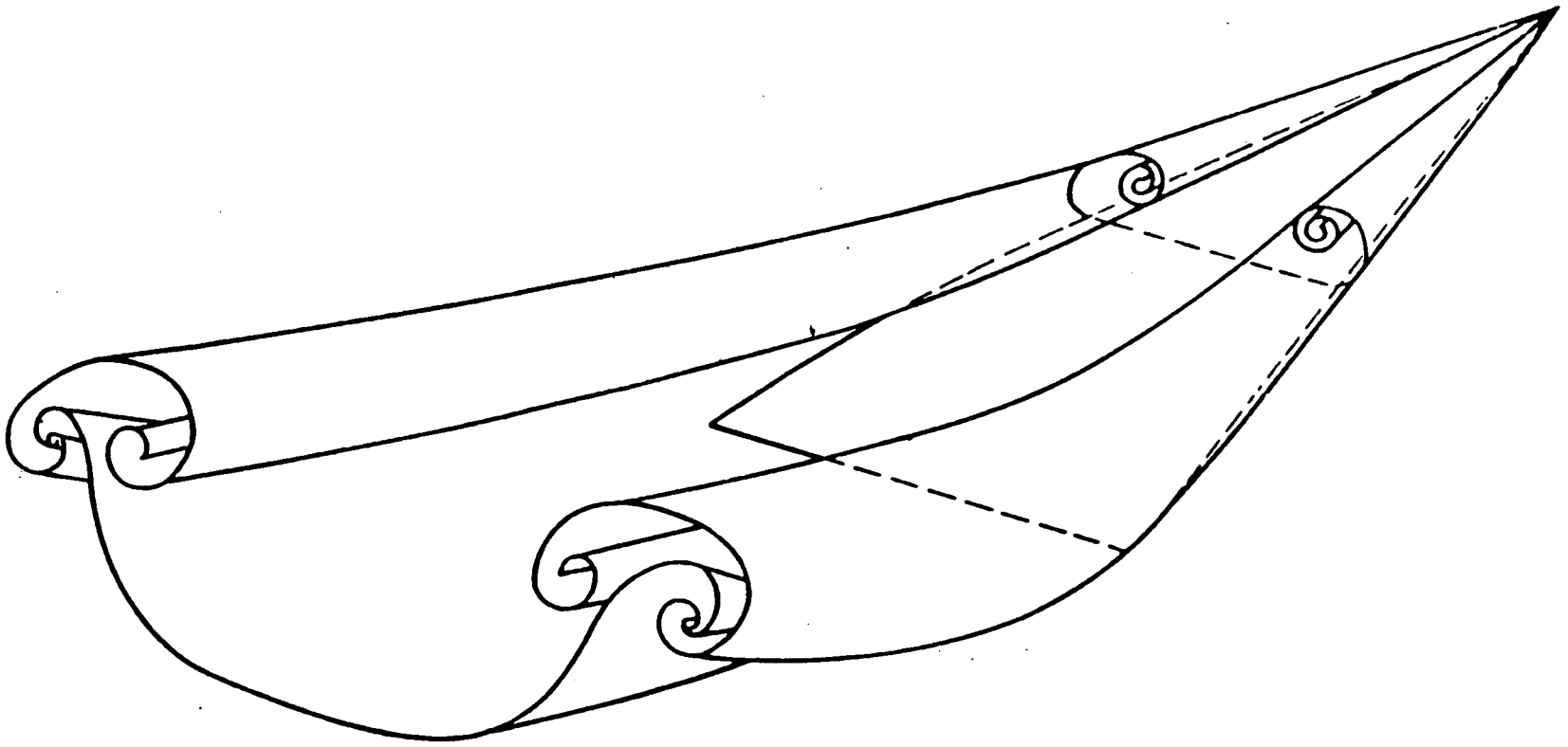
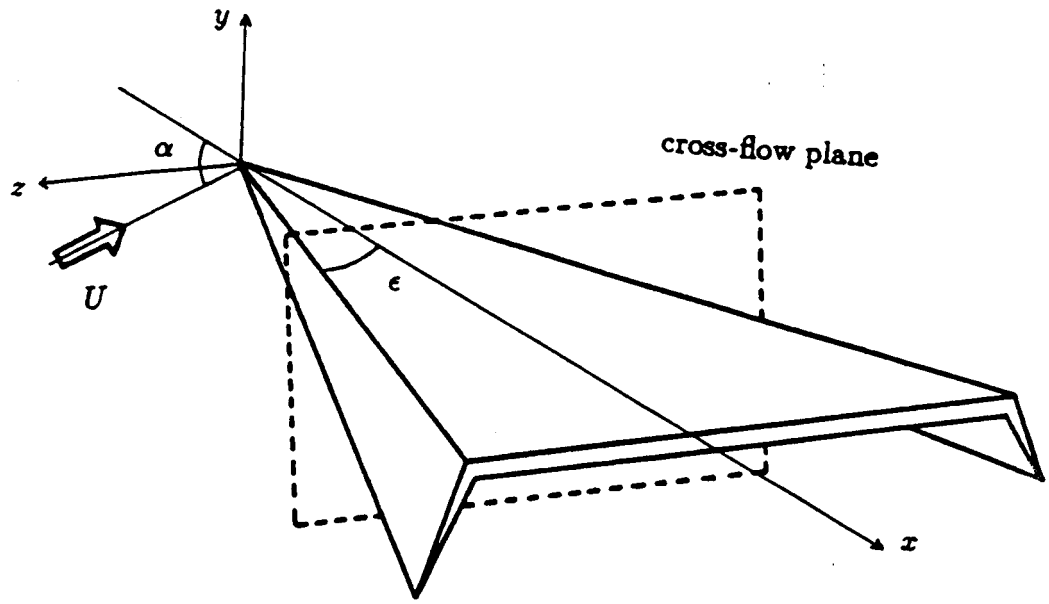
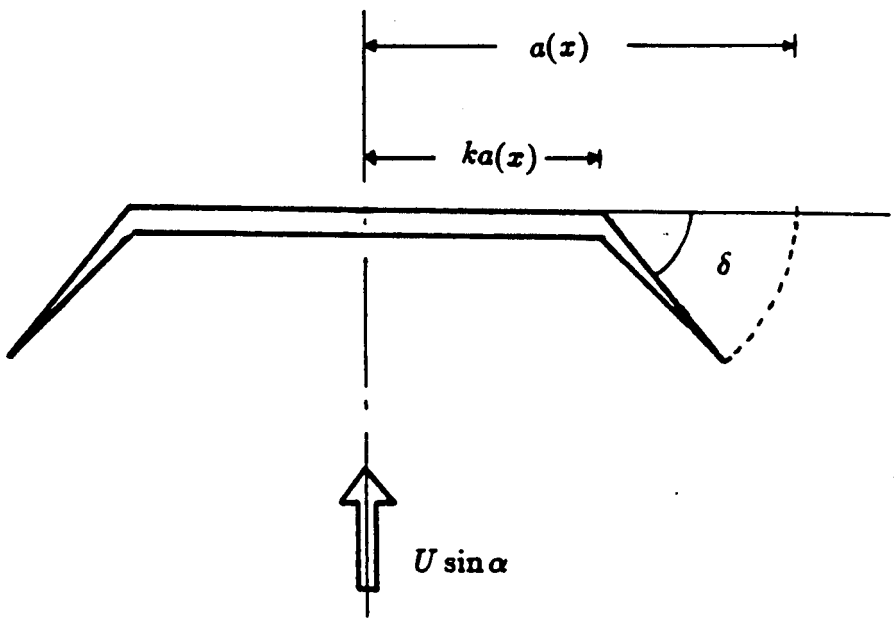


Fig. 1 Overall flow field past a delta wing



(a) Delta wing coordinate system



(b) Cross-flow plane

Fig. 2

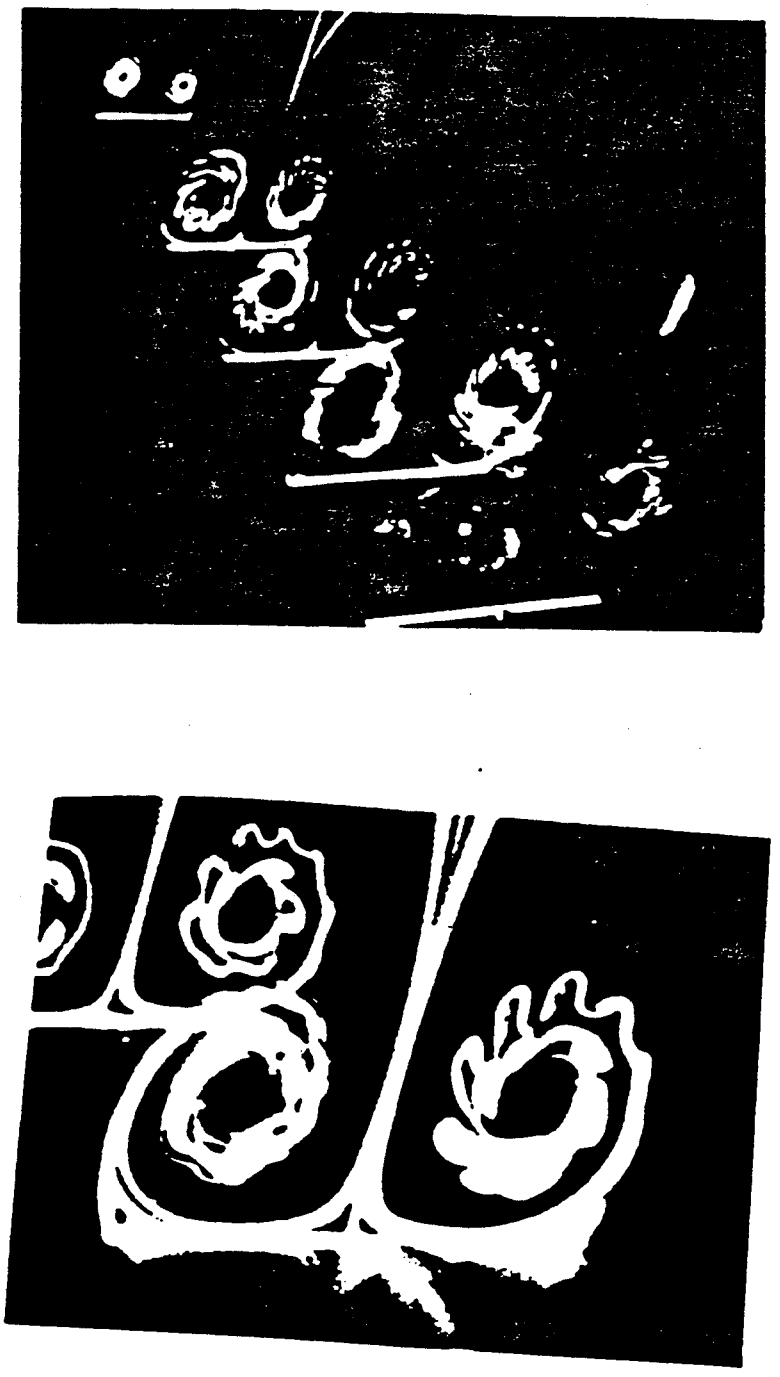
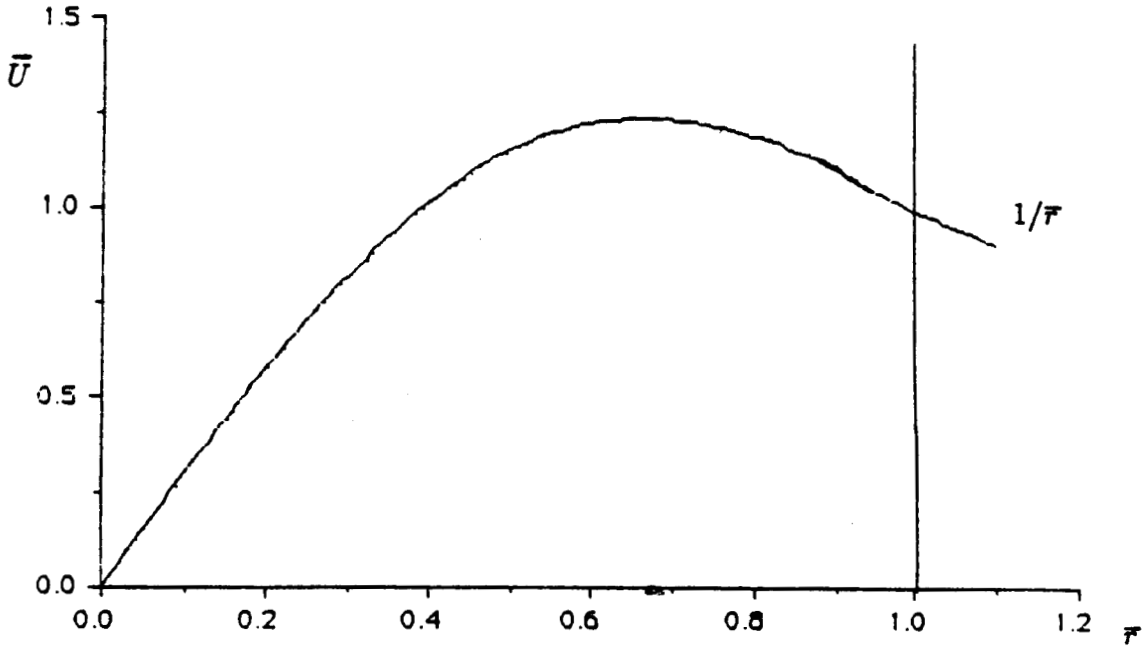
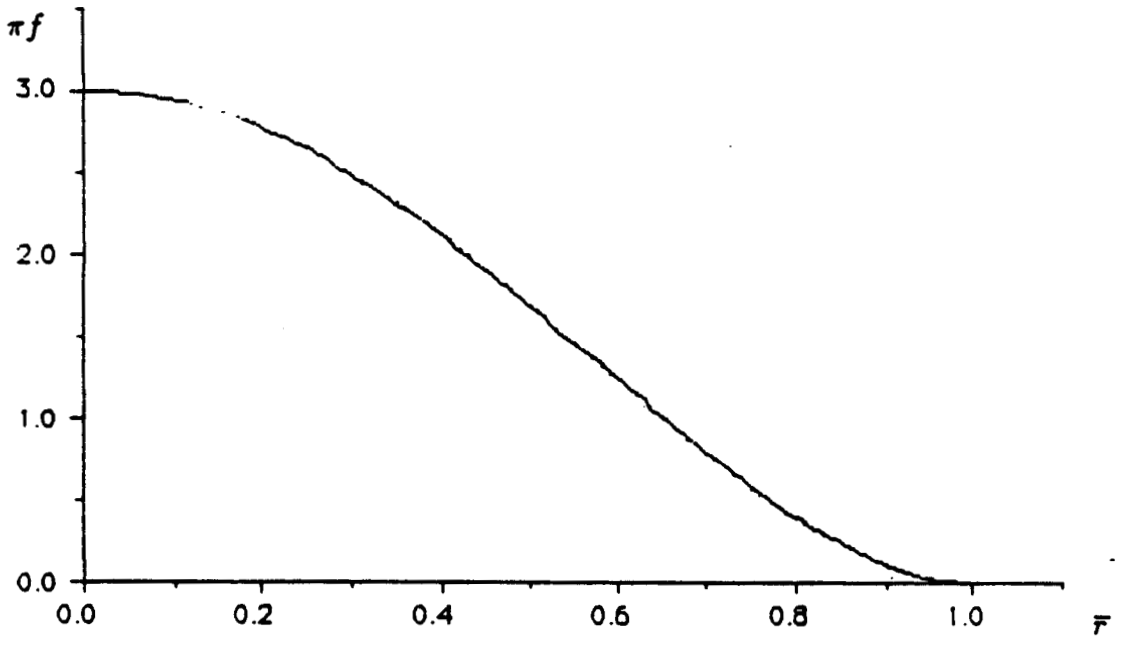


Fig. 3 Experimental visualization of the rolled-up shear layer, taken from Ref. 25.

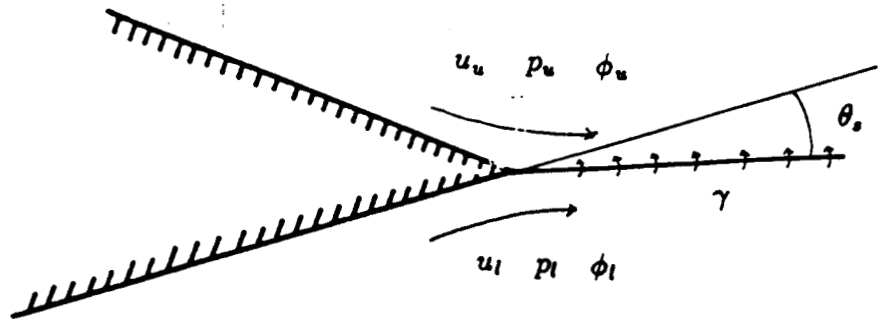


(a) Velocity distribution

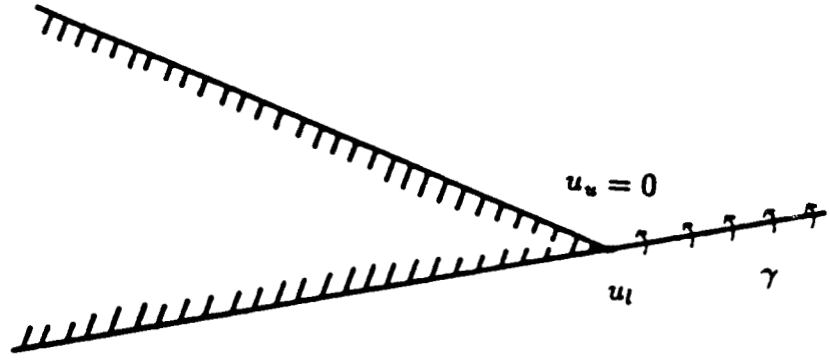


(b) Vorticity distribution

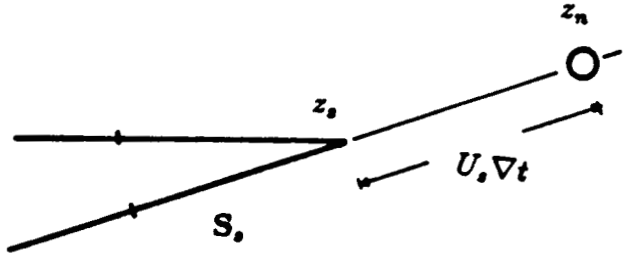
Fig. 4 Spalart's vortex core model



(a) General case



(b) Fixing separation angle, $\theta_s = 0$



(c) New vortex position

Fig. 5 Flow around separation point

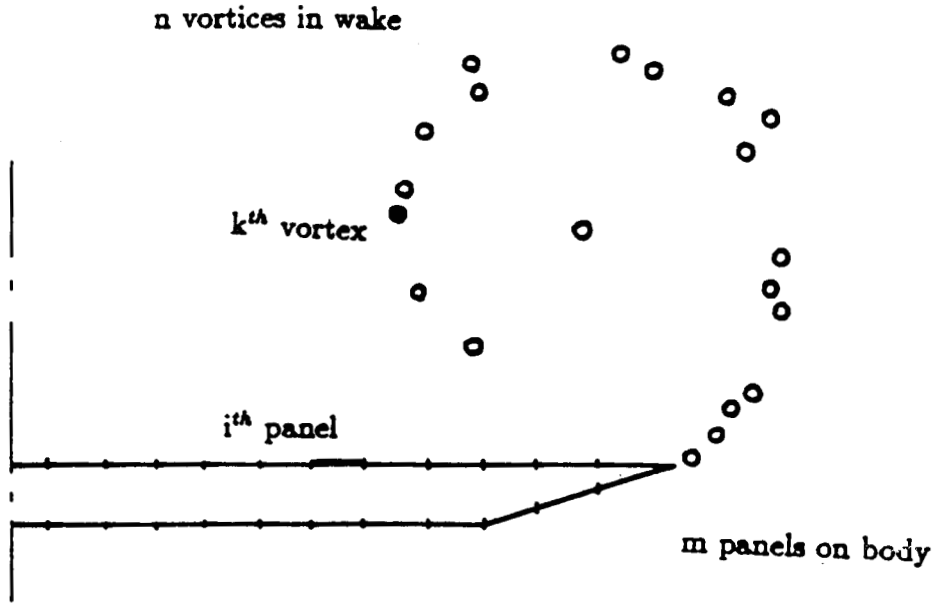


Fig. 6 Body panels and wake vortices

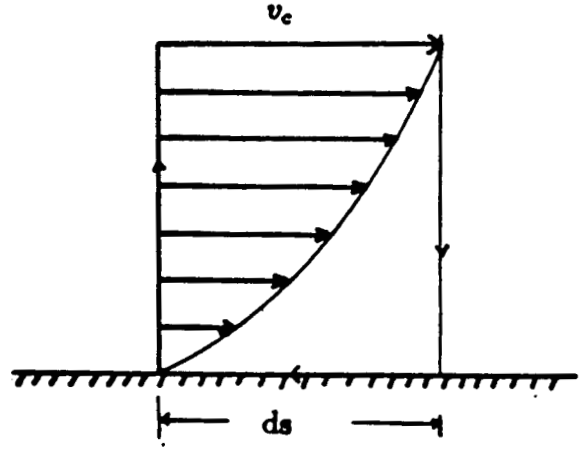
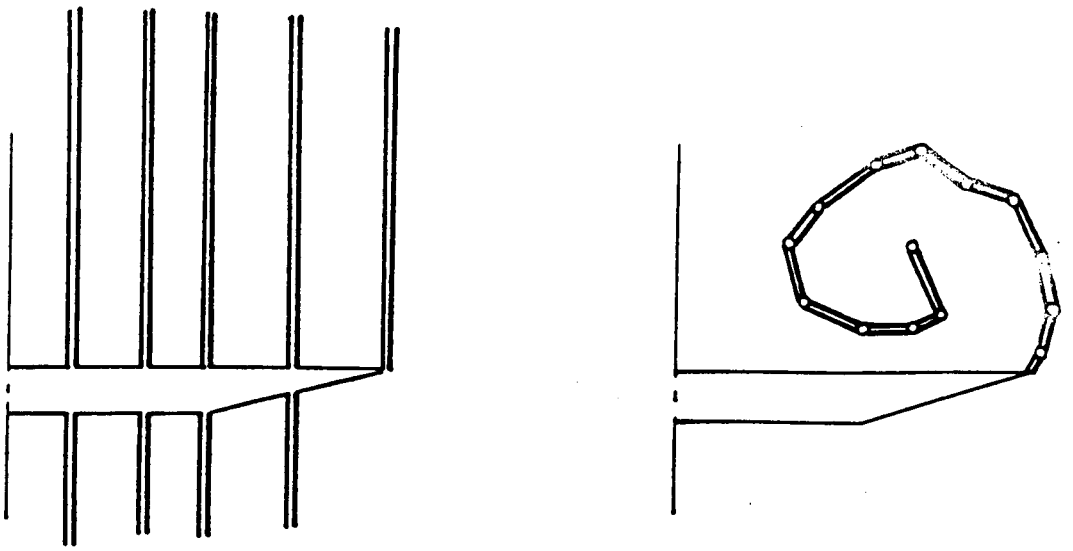


Fig.7 Boundary layer and outer velocity



(a) For bound vortices

(b) For wake vortices

Fig. 8 Branch lines

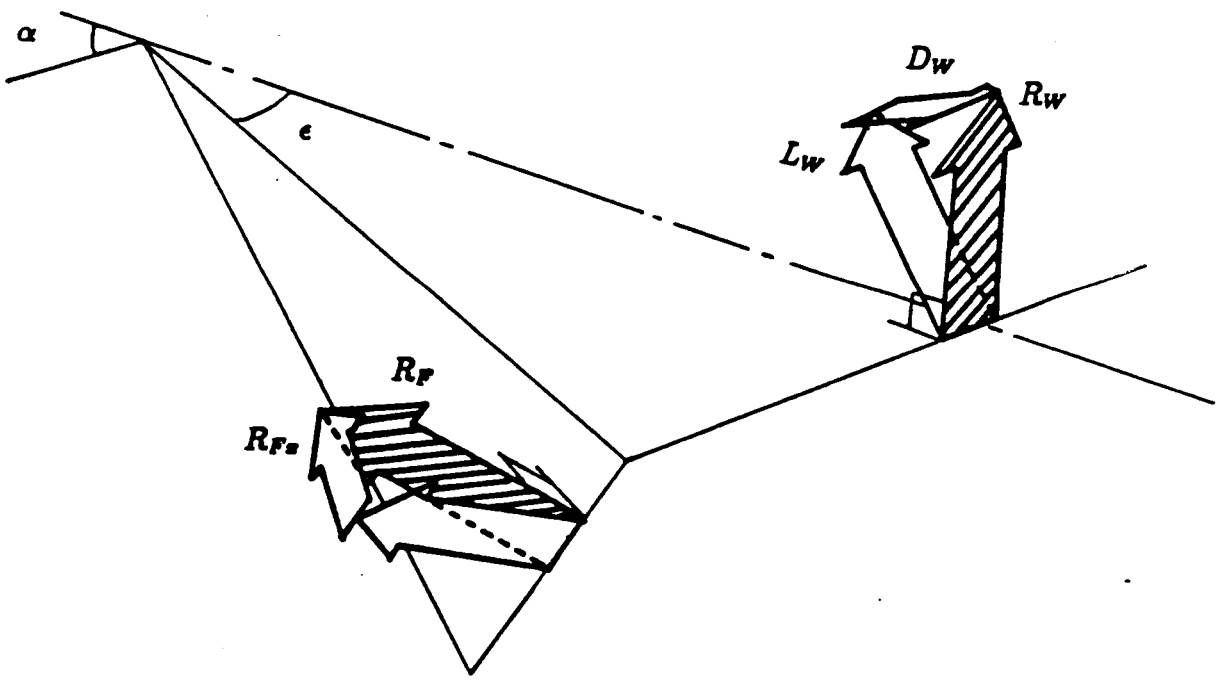
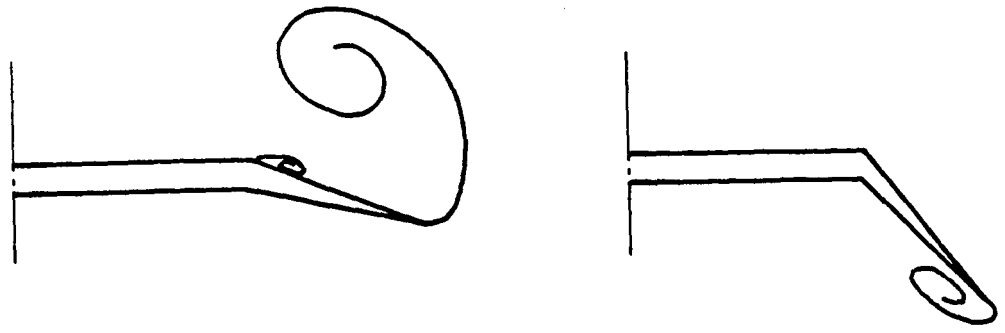


Fig. 9 Force components on wing and flap surfaces



(a) below the limit flap angle (b) above the limit flap angle

Fig. 10 Vortical flow for various flap angles

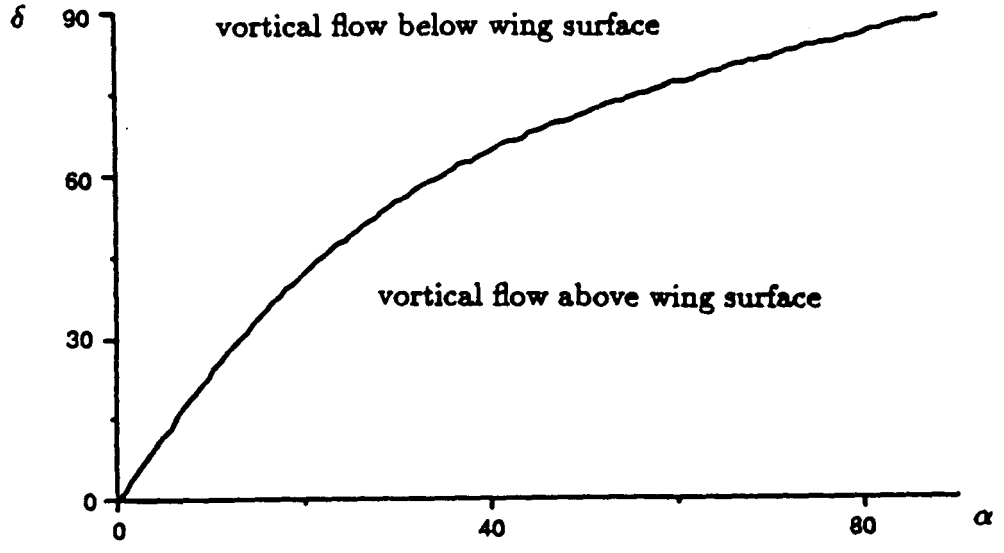


Fig. 11 Boundary of the flap angle

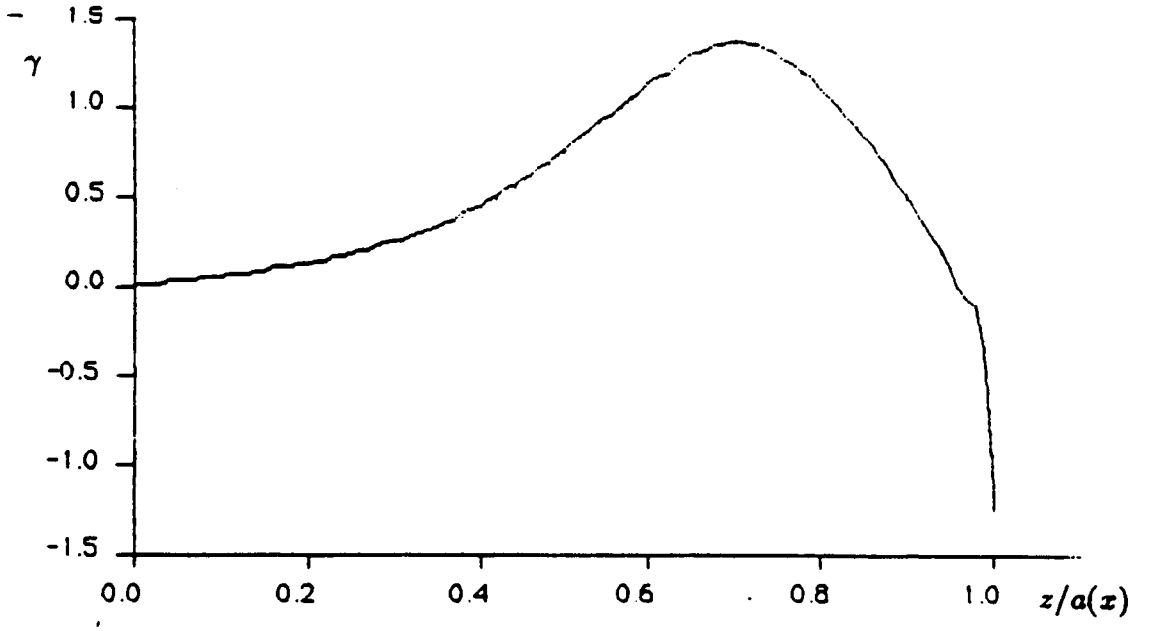


Fig. 12 Chordwise bound vorticity distribution

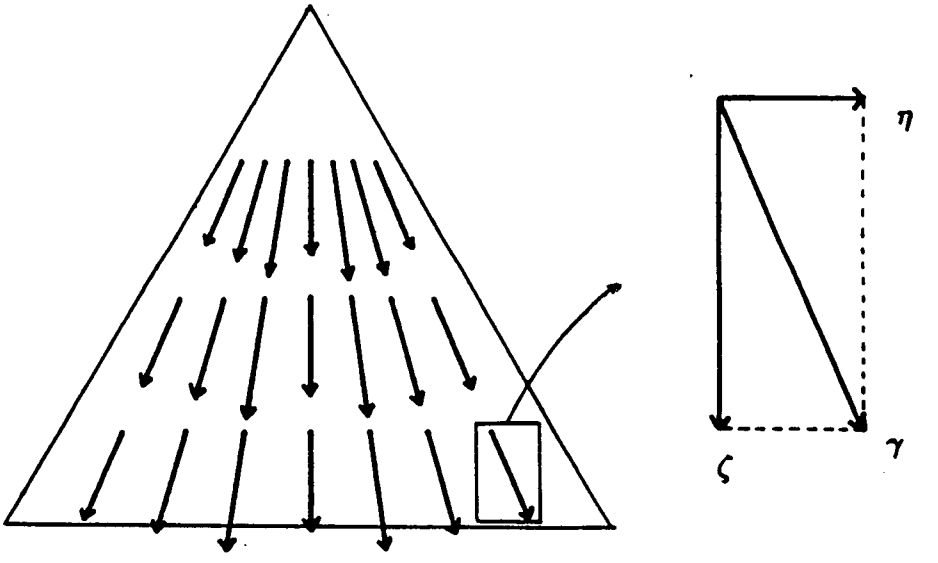
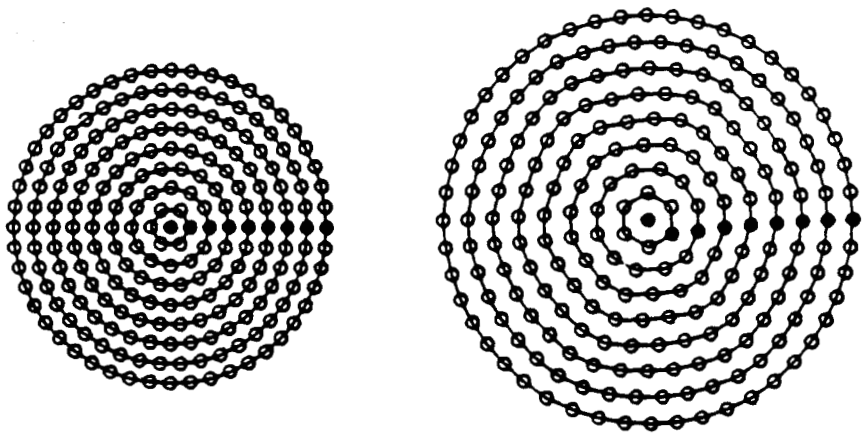


Fig. 13 Bound vorticity components



(a) initial state (b) after some time steps

Fig. 14 Numerical diffusion illustrated by discrete simulation of Rankine vortex, taken from Ref. 36.

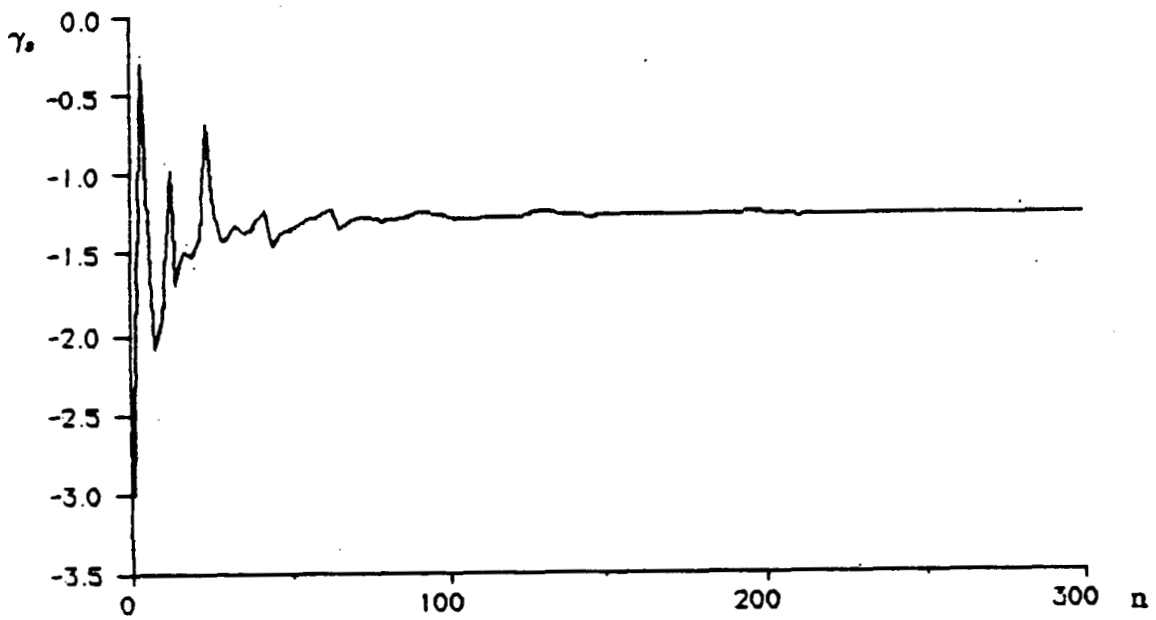
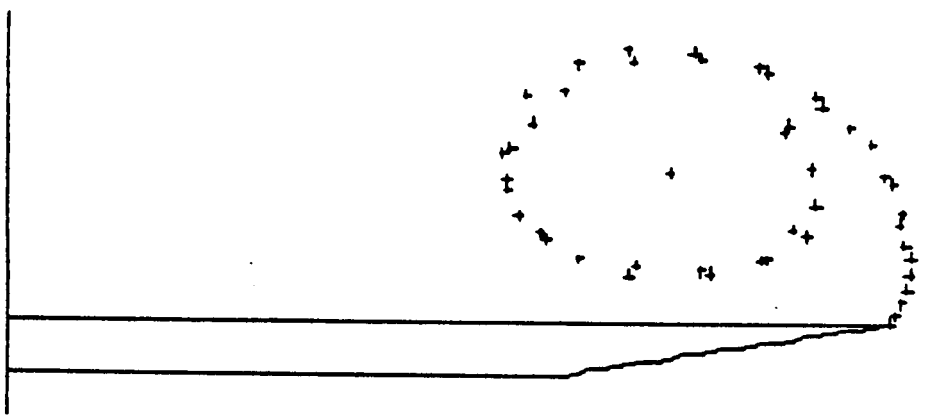


Fig. 15 Convergence of vortex strength at the separation point, initial guess is $3U_\infty$

(a) Rolled-up shear layer without core model



(b) Rolled-up shear layer with core model, rotation angle is 2.5π

Fig. 16 Effect of core model on shear layer

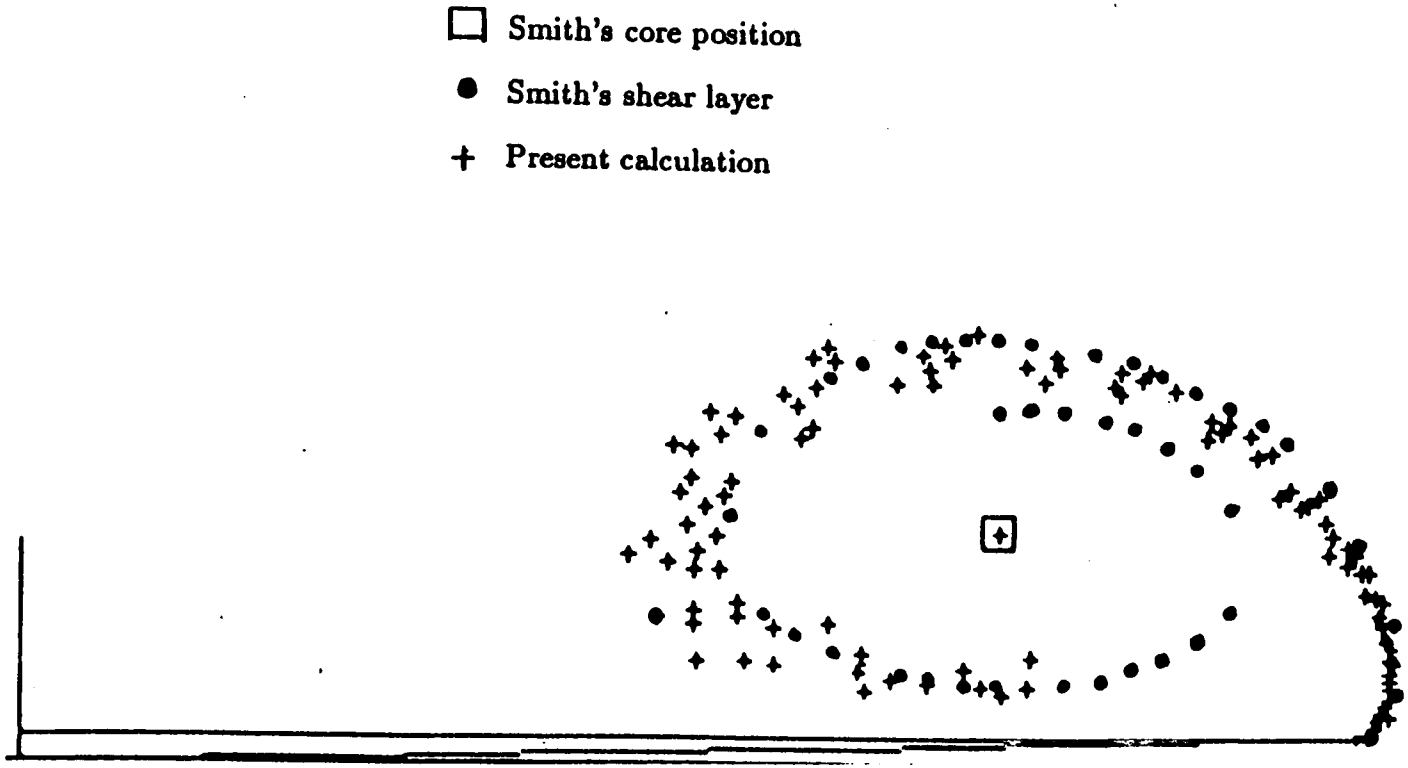
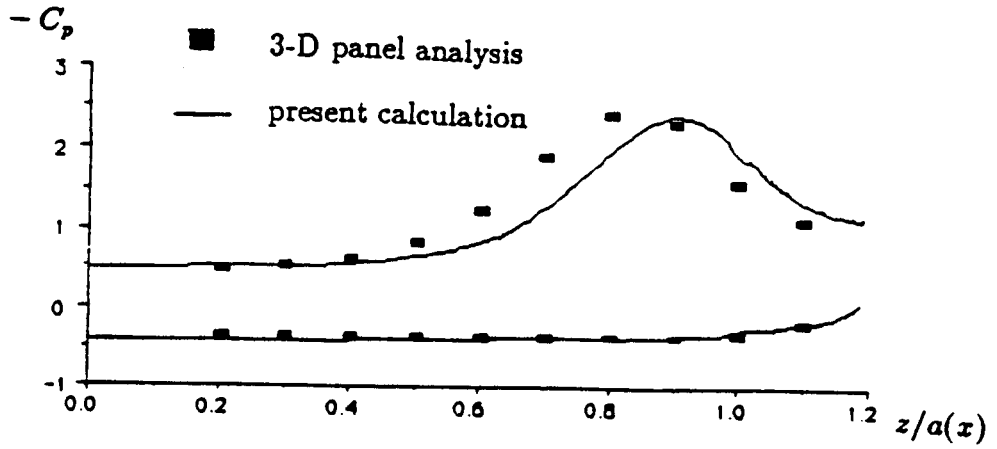
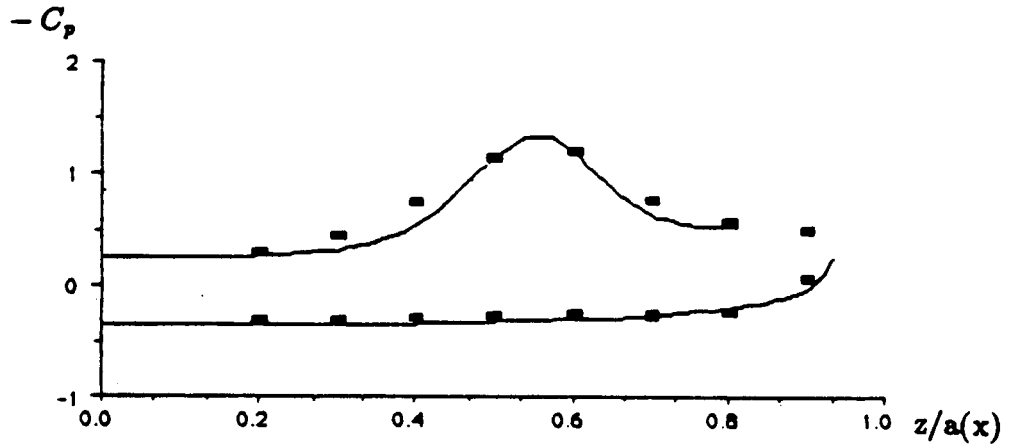


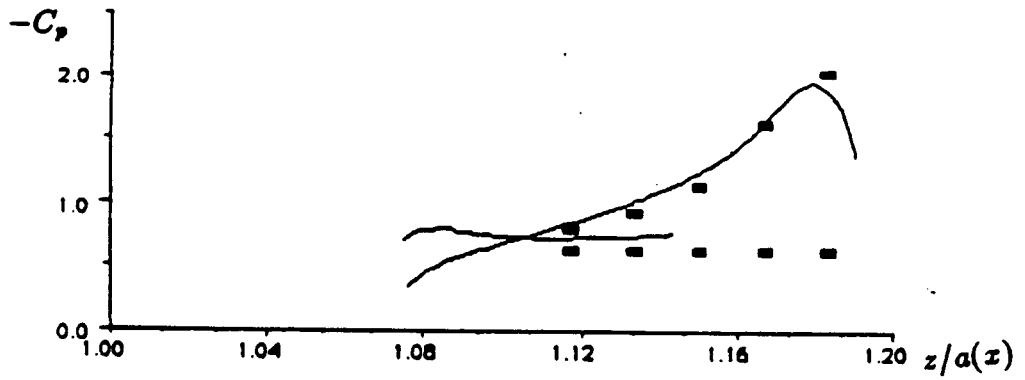
Fig. 17 Comparison with Smith's calculation for $\alpha/\epsilon = 1$



(a) no fence

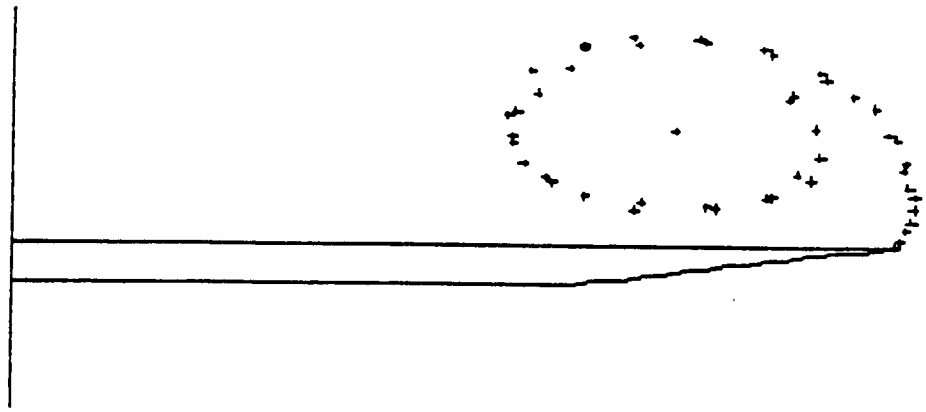


(b) main wing surface for $\delta = -130$ deg.

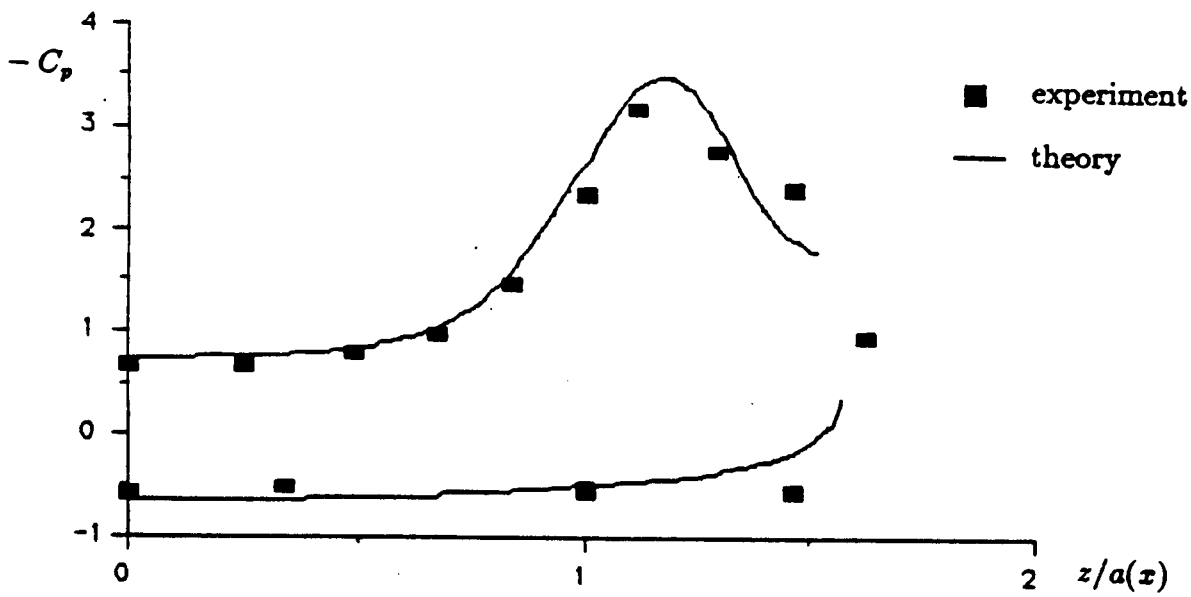


(c) fence surface for $\delta = -130$ deg.

Fig. 18 Comparison with 3-D panel analysis

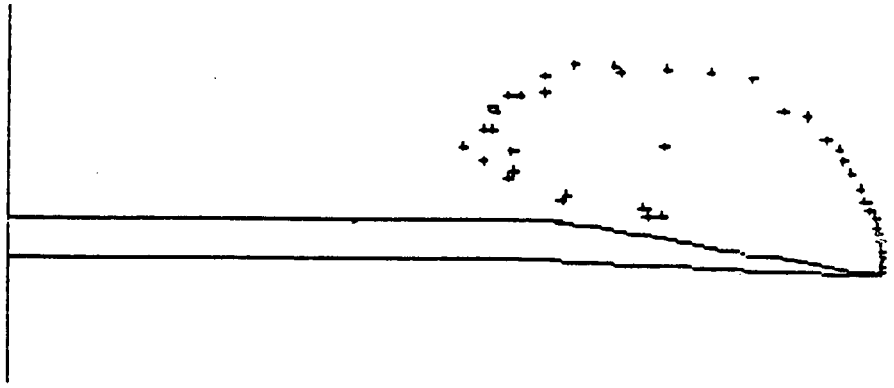


(a) Rolled-up shear layer

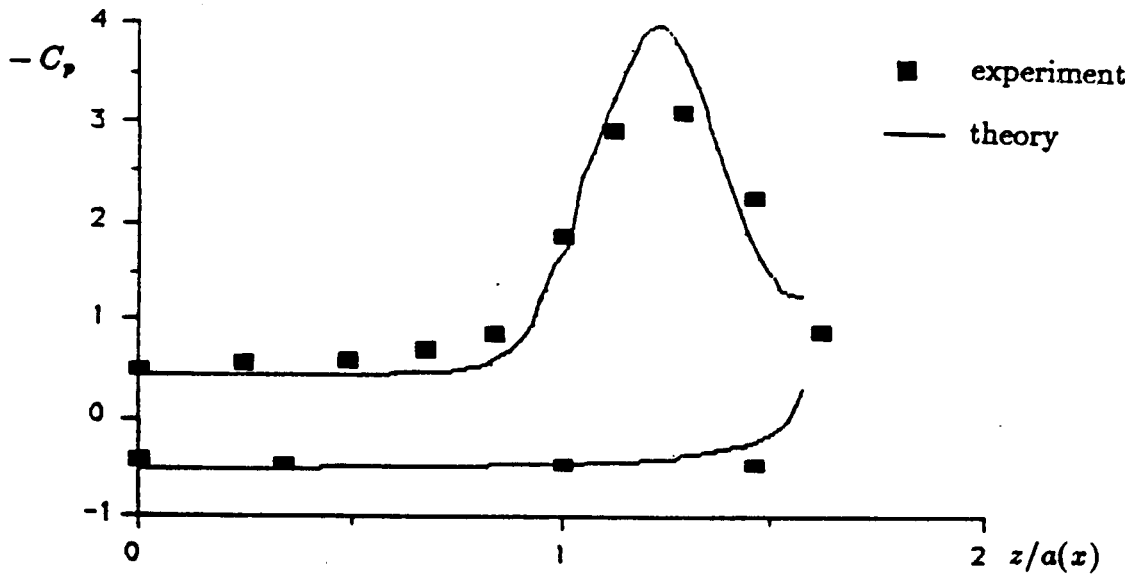


(b) Surface pressure distribution

Fig. 19 α = 25 deg., δ = 0 deg.

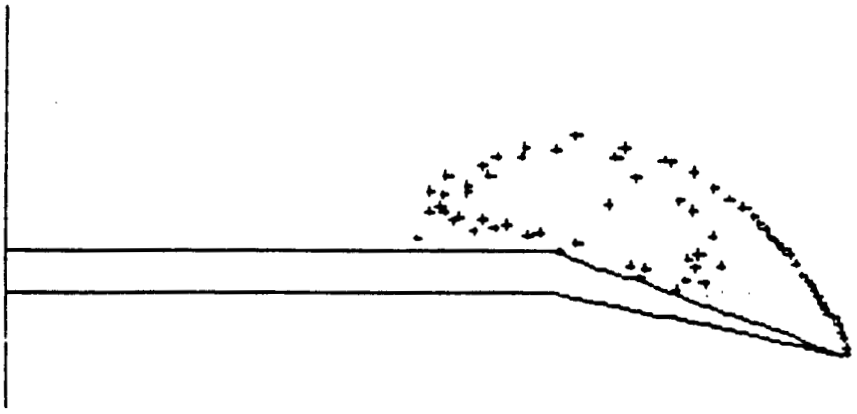


(a) Rolled-up shear layer

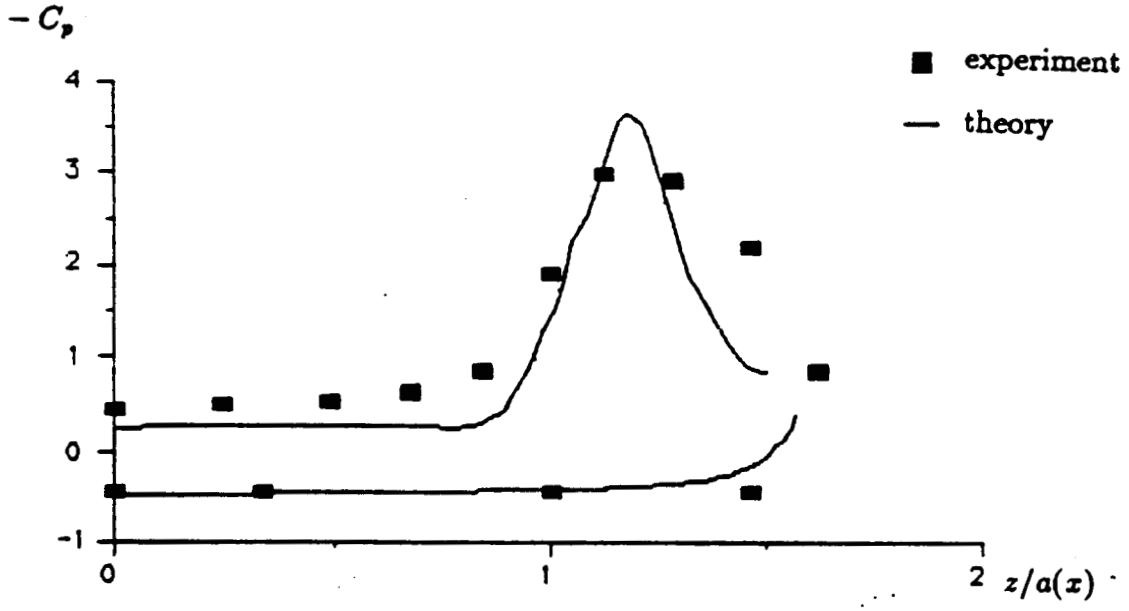


(b) Surface pressure distribution

Fig. 20 α = 25 deg., δ = 15 deg.

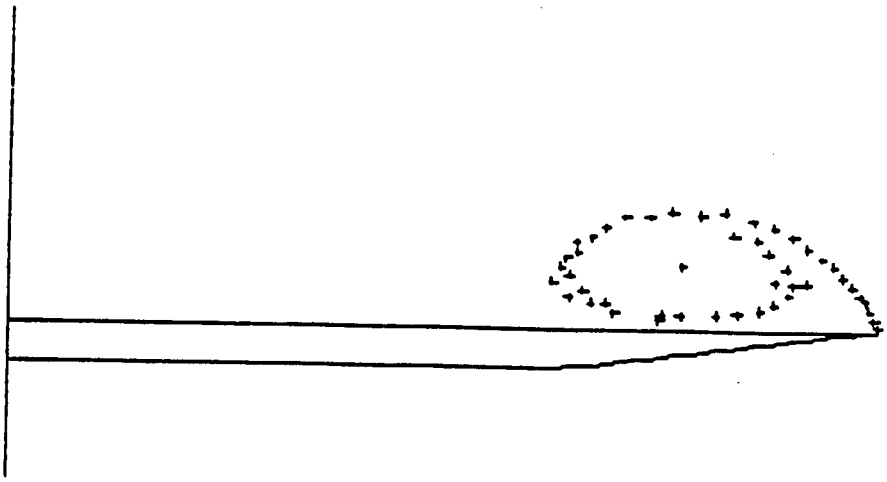


(a) Rolled-up shear layer

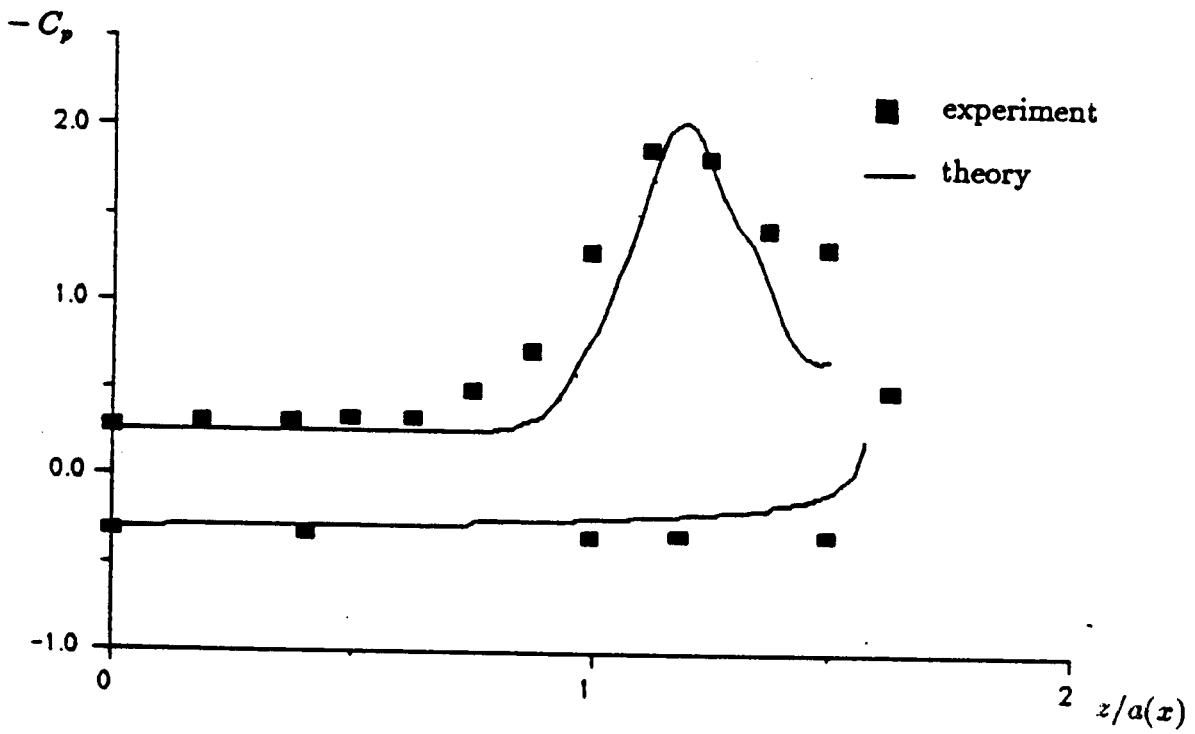


(b) Surface pressure distribution

Fig. 21 α = 25 deg., δ = 30 deg.

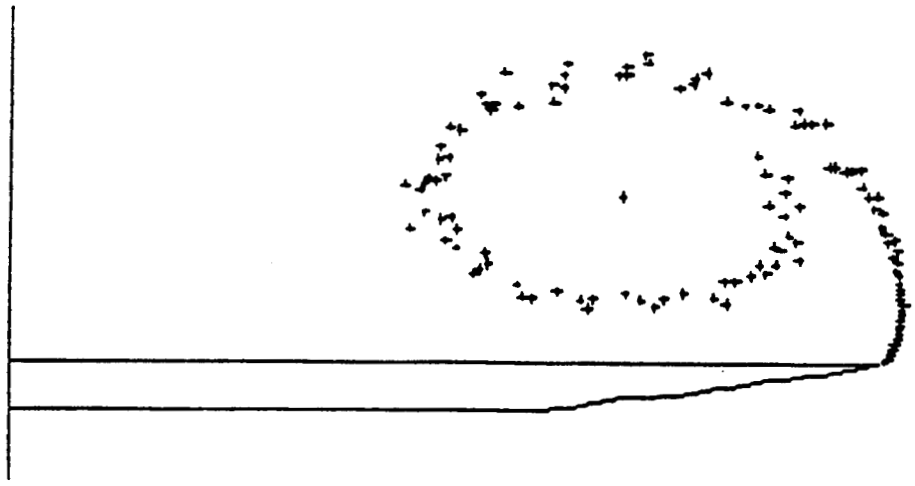


(a) Rolled-up shear layer

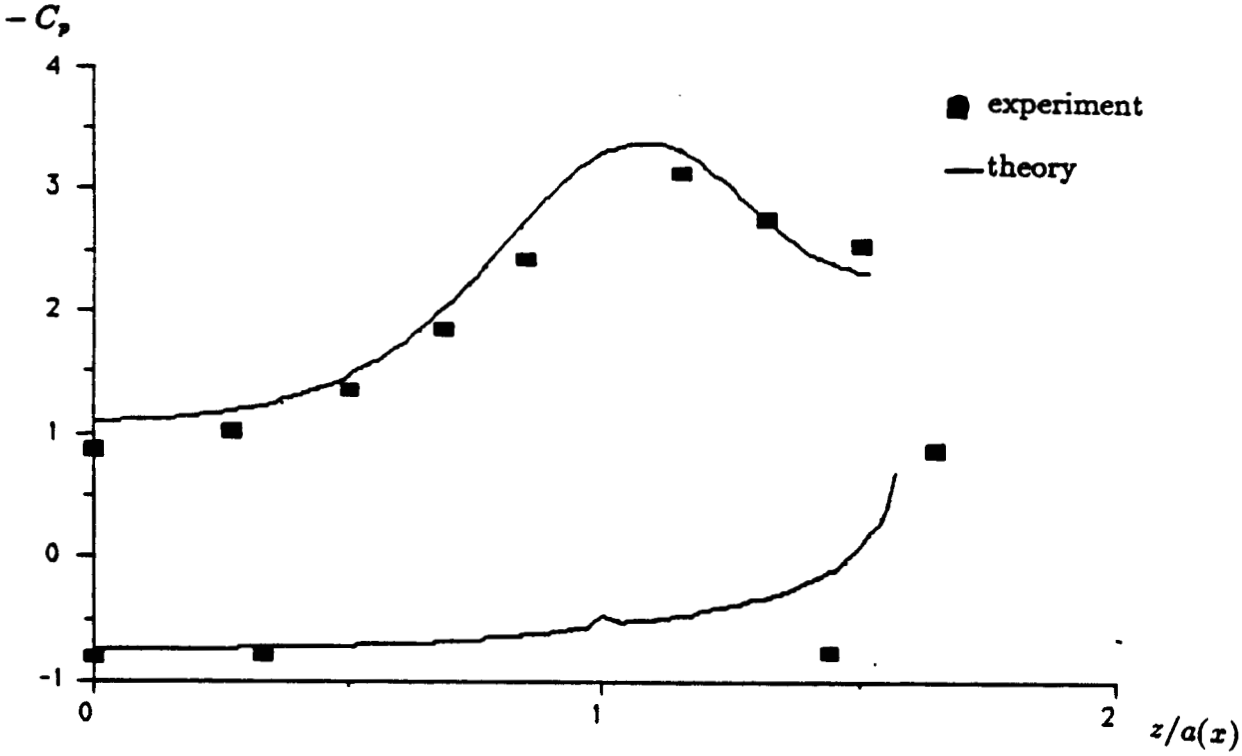


(b) Surface pressure distribution

Fig. 22 α = 15 deg., δ = 0 deg.

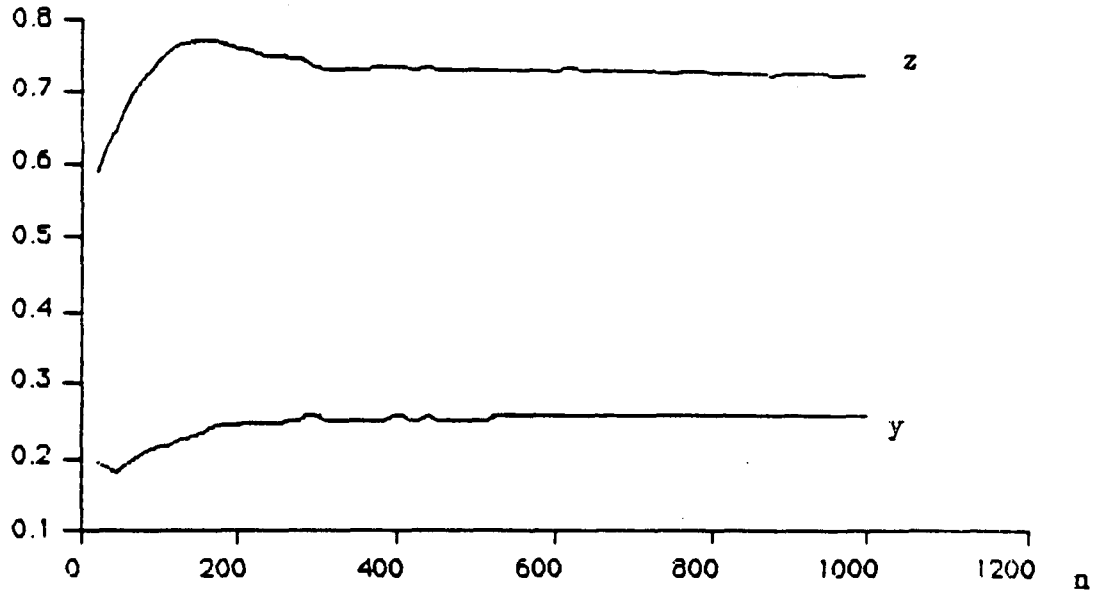


(a) Rolled-up shear layer

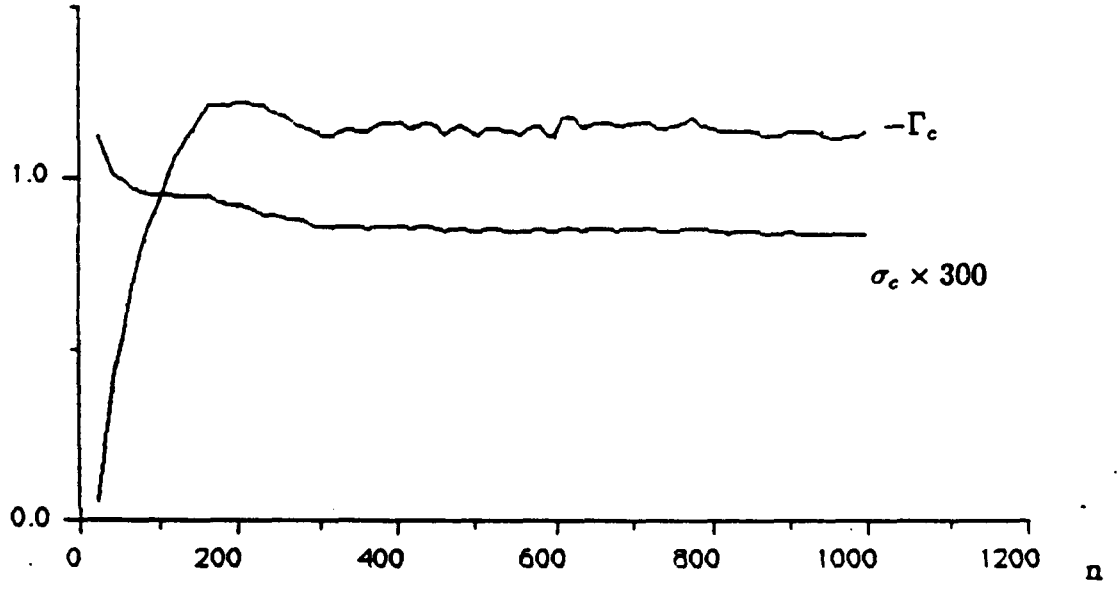


(b) Surface pressure distribution

Fig. 23 α = 35 deg., δ = 0 deg.

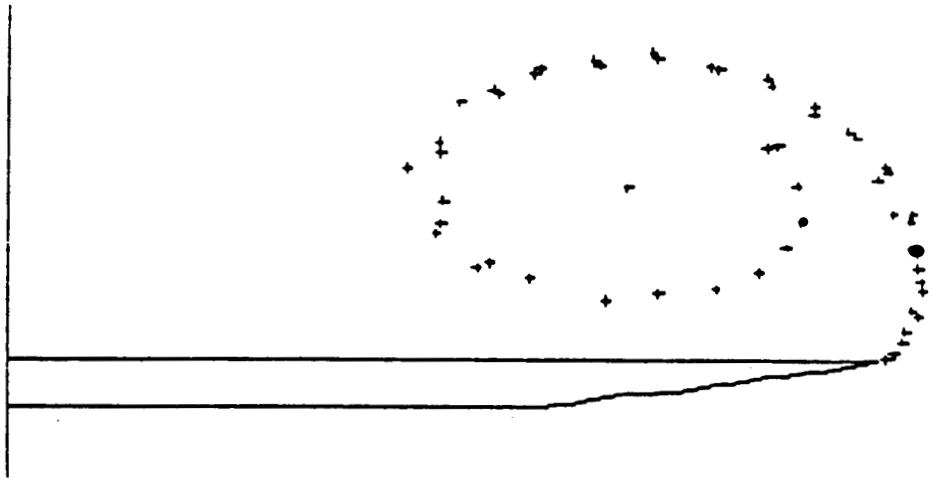


(a) Core position

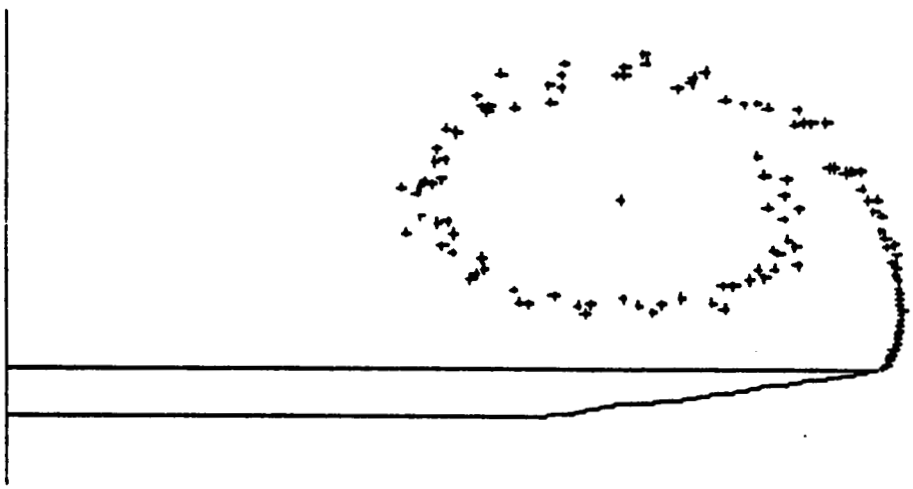


(b) Core strength and core radius

Fig. 24 Convergence of calculation

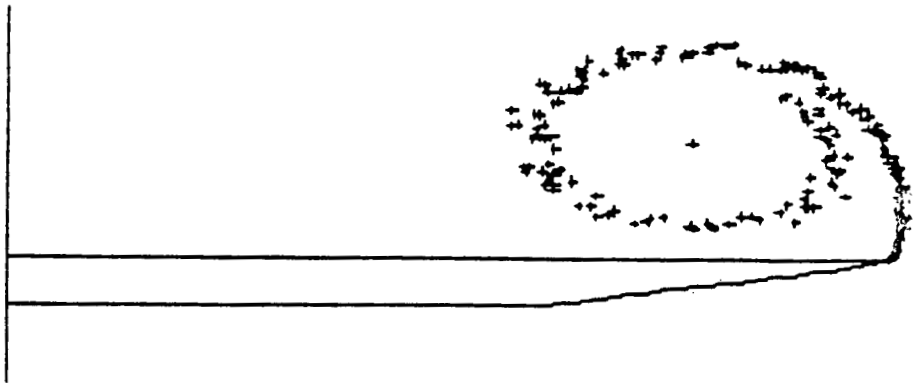


(a) 4^{th} order Runge-Kutta method

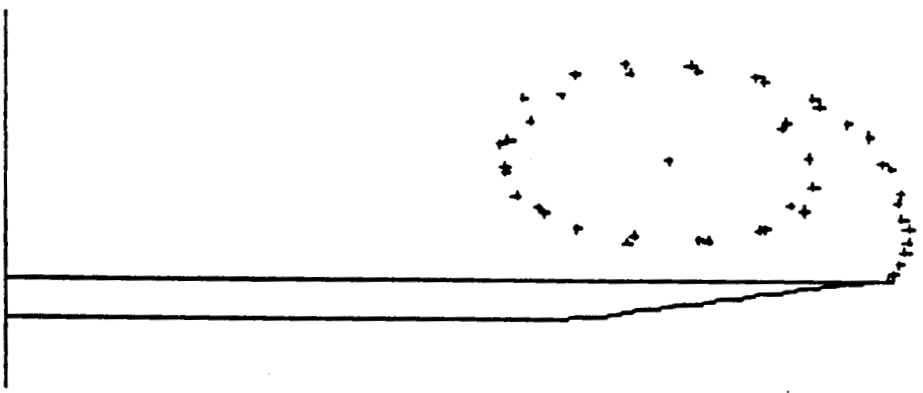


(b) 1^{th} order Euler method

Fig. 25 Effect so integration scheme on shear layer structure

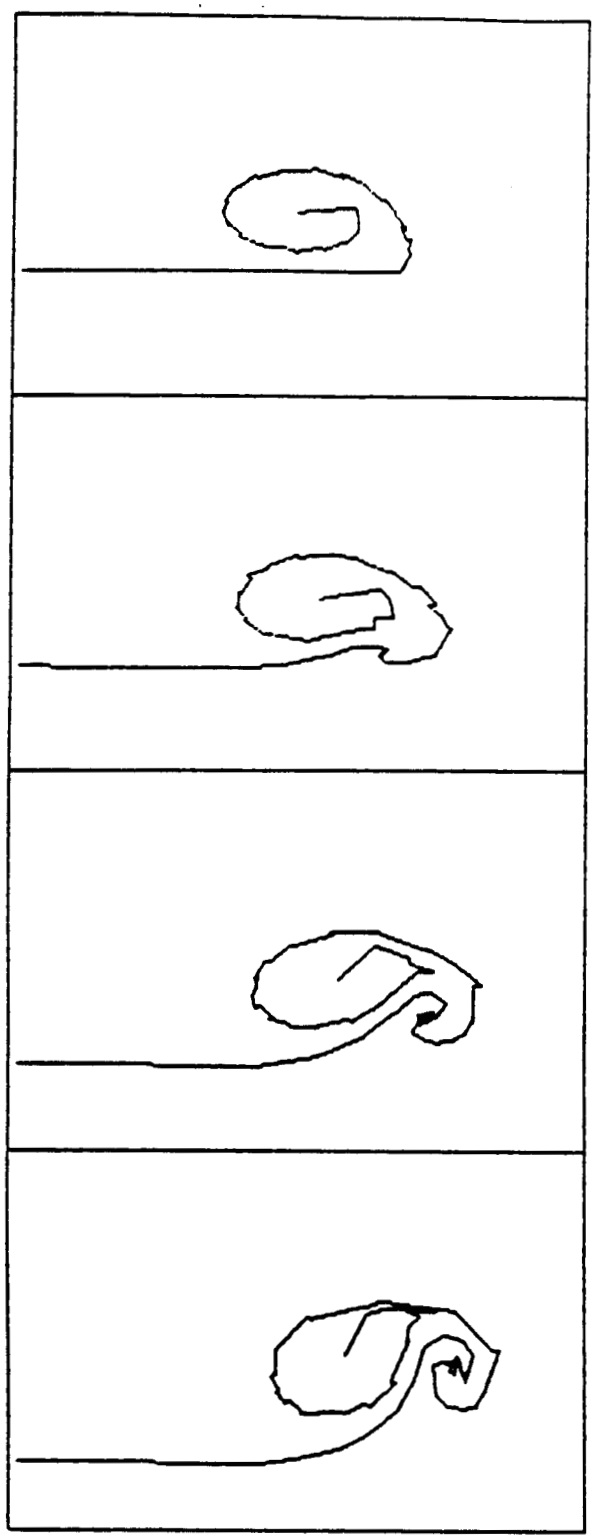


(a) no merging between neighboring vortices

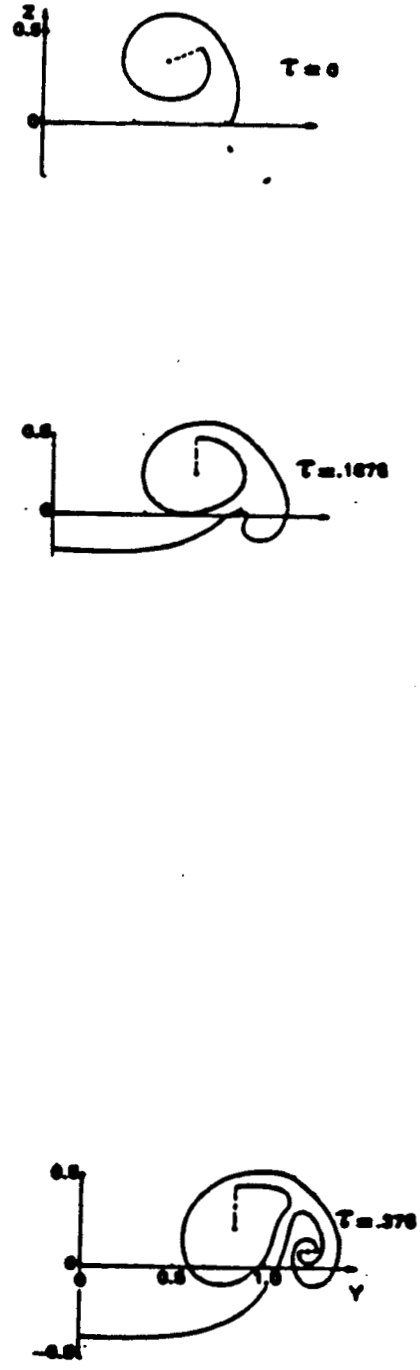


(b) merging between neighboring vortices

Fig. 26 Effect of merging on shear layer structure

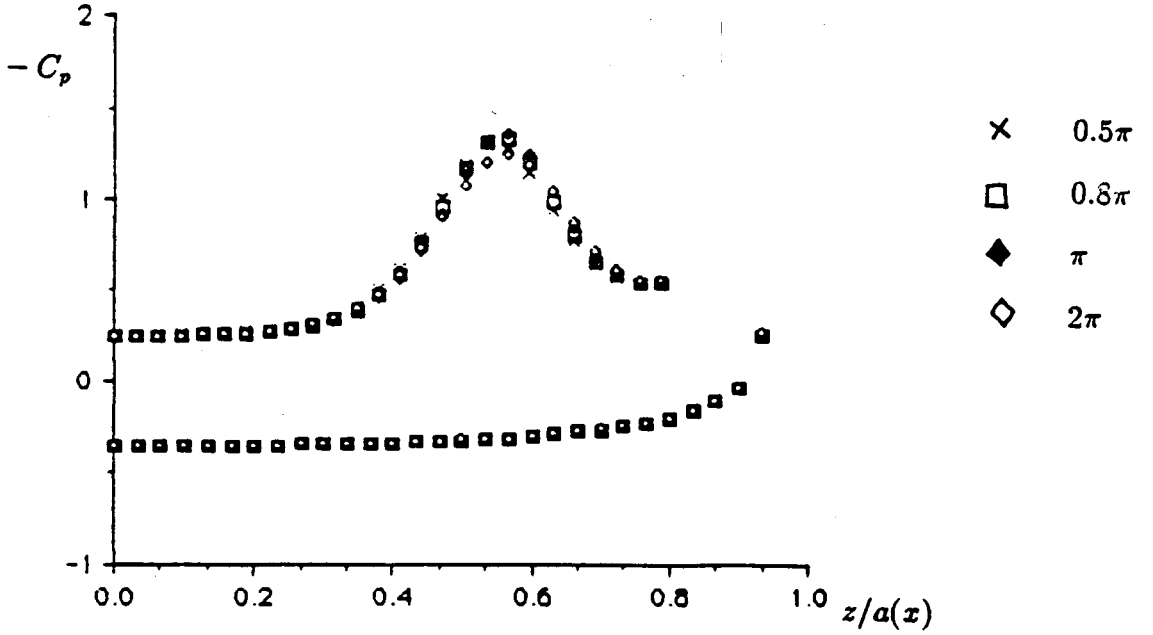


(a) Present calculation

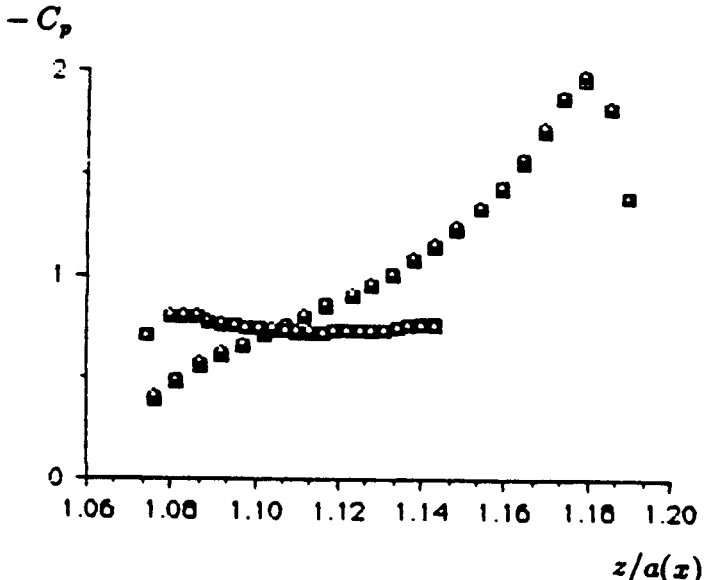


(b) Higher-order panel method

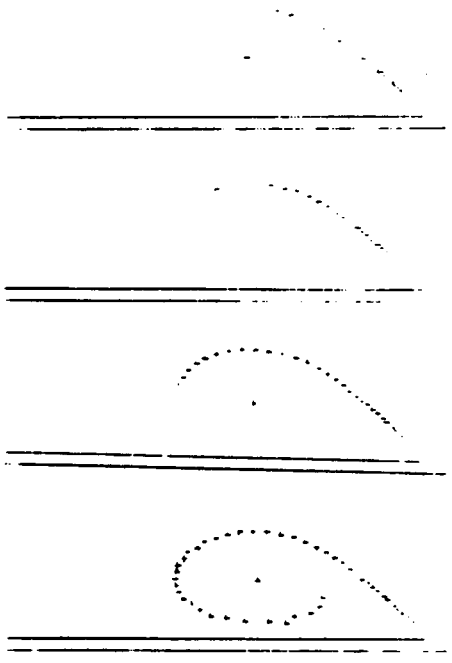
Fig. 27 Trailing edge wake evolution



(a) main wing surface



(b) fence surface



(c) rolled-up shear layer

Fig. 28 The effect of core rotation angle

APPENDIX

Fortran Program

```
C*************************************************************************

C  VORTEX PANEL METHOD FOR FLAT DELTA WING WITH FLAP IN CONICAL FLOW
C             MODIFIED FOR SYMMETRIC CASE

C*************************************************************************

      COMMON/OPTION/INSTEP,NSTEP,NPLT,IFPL,IFFL,UMAX

C   CONSTRUCTING BODY SHAPE AND GEOMETRIC INFLUENC MATRIX  *  *  *
      CALL INITIAL
C

C   IFFL=0 : NEW CASE,     IFFL=1 : CONTINUEING CASE    ----------
      IF(IFFL.EQ.0) GO TO 111
      CALL INFO

111   NNSTEP=NSTEP-30

      DO 100 ISTEP=INSTEP,NSTEP
        JJ=ISTEP

C   MATRIX INVERSION WITH THE INFLUENCE OF THE WAKE  *  *  *
        CALL SOLVE
C

C   FOR NEXT TIME STEP  *  *  *
        CALL MOVE(JJ)
C

C   IFPL=1 : NO PLOT,      IFPL=2 : PLOT    ----------------------
        IF(IFPL.EQ.1) GO TO 10
        IF(MOD(ISTEP,NPLT).EQ.0) CALL VXPLOT
        IF(ISTEP.LT.NNSTEP) GO TO 10
        CALL PRESSURE(ISTEP)
100   CONTINUE

C   CALCULATION OF LIFT  *  *  *
      CALL LIFT
C

C   FOR PUTPUT PRINT  *  *  *
      CALL OUTPUT
C

C   FOR NEXT CONTINUING CASE  *  *  *
      CALL RESER
C

      STOP
      END


C   ===========================================================
      SUBROUTINE OUTPUT

      IMPLICIT COMPLEX Z

      COMMON/CORE/VAL1,VAL2
      COMMON/MATRIX/NBC,AA(150,150),AX(150)
      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,DELTA
      COMMON/OPTION/INSTEP,NSTEP,NPLT,IFPL,IFFL,UMAX
      COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)
      COMMON/PRESSURE/CPL(150),CPU(150),CPML(150),CPMU(150)
      COMMON/SVEL/XUL(150),XUU(150),XVL(150),XVU(150),
```

```
     *            XWL(150),XWU(150)
      COMMON/LIFT/XLFTW,XLFTF,XDRGW,XDRGF,XLIFT,XDRAG,RATIO

      ALPHD=ALPH*180./PI
      EPSLD=EPSL*180./PI
      DELTAD=DELTA*180./PI

      WRITE(40,100) ALPHD,EPSLD,DELTAD,DX,NSTEP,X0,VAL1,VAL2
100   FORMAT(///10X,'ANGLE OF ATTACK =    ',F10.4/
     *          10X,'HALF APEX ANGLE =    ',F10.4/
     *          10X,'FLAP ANGLE      =    ',F10.4/
     *          10X,'DX              =    ',F10.4/
     *          10X,'NSTEP           =    ',I5/
     *          10X,'INTIAL SPAN POINT = ',F10.4/
     *          10X,'CORE ANGLE IN RAD. =',F5.2,5X,F5.2)

      RRATIO=1./TAN(ALPH)
      WRITE(40,3000)XLFTW,XLFTF,XDRGW,XDRGF,XLIFT,XDRAG,RATIO,RRATIO
3000  FORMAT(///10X,'LIFT ON WING  =  ',E10.3/
     1          10X,'LIFT ON FLAP  =  ',E10.3/
     2          10X,'DRAG ON WING  =  ',E10.3/
     3          10X,'DRAG ON FLAP  =  ',E10.3/
     4          10X,'TOTAL LIFT    =  ',E10.3/
     5          10X,'TOTAL DRAG    =  ',E10.3/
     6          10X,'LIFT-TO-DRAG RATIO   =  ',E10.3/
     7          10X,'LIFT-TO DRAG-RATIO W/O FLAP   =   ',E10.3////)


      WRITE(40,1100)
1100  FORMAT(10X,'PANEL NUMBER AND BOUND VORTEX STENGTH'/)
      DO 1200 I=1,NBODYH
1200  WRITE(40,1300)I,AX(I),CPL(I),XUL(I),XVL(I),XWL(I),
     *              CPU(I),XUU(I),XVU(I),XWU(I)
1300  FORMAT(3X,I3,E10.3,3X,4(E10.3,3X)/19X,4(E10.3,3X))

      DO 300 I=1,NBODYH
      CPP=-CP(I)
300   WRITE(50,888)CPP
888   FORMAT(5X,F10.5)

      WRITE(40,500)
500   FORMAT(///10X,'VORTEX STRENGTH AND POSITION FOR LEADING EDGE'/)
      DO 600 I=1,NWAKEL
      IWL=2*I-1
600   WRITE(40,700) IWL,GWAKE(IWL),ZWAKE(IWL),WCORE(IWL)
700   FORMAT(5X,I5,5X,E10.3,5X,2E10.3,5X,E10.3)

      WRITE(40,1000)
1000  FORMAT(///10X,'VORTEX STRENGTH AND POSITION FOR HINGE VORTEX'/)
      DO 800 I=1,NWAKEF
      IWF=2*I
800   WRITE(40,900) IWF,GWAKE(IWF),ZWAKE(IWF),WCORE(IWF)
900   FORMAT(5X,I5,E10.3,5X,2E10.3,5X,E10.3)

      RETURN
      END



C     ================================================================

      SUBROUTINE RESER

      IMPLICIT COMPLEX Z
```

```
      COMMON/SEPRAT/ISEP(2),ZSSP(2)
      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,DELTA
      COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)
      COMMON/MATRIX/NBC,AA(150,150),AX(150)
      COMMON/BODY/M1,M2,M3,M4

      IWL=2*NWAKEL-1
      IWF=2*NWAKEF
      OPEN(UNIT=11,FILE='NFILE',FORM='UNFORMATTED',STATUS='NEW')
      WRITE(11) X,XK,TGBD,NBODY,NBODYH,NBC,NWAKEL,NWAKEF
      WRITE(11) M1,M2,M3,M4
      WRITE(11) (ISEP(I),ZSSP(I),I=1,2)
      WRITE(11) (ZBODY(I),ZSBD(I),ALBD(I),I=1,NBODY)
      WRITE(11) ((AA(I,J),J=1,150),AX(I),I=1,150)
      WRITE(11) (GWAKE(I),ZWAKE(I),WCORE(I),I=1,IWL,2)
      WRITE(11) (GWAKE(I),ZWAKE(I),WCORE(I),I=2,IWF,2)
      CLOSE(UNIT=11)

      RETURN
      END


C     ================================================================

      SUBROUTINE INFO

      IMPLICIT COMPLEX Z

      COMMON/SEPRAT/ISEP(2),ZSSP(2)
      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,DELTA
      COMMON/BODY/M1,M2,M3,M4
      COMMON/MATRIX/NBC,AA(150,150),AX(150)
      COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)

      OPEN(UNIT=11,FILE='NFILE',FORM='UNFORMATTED',STATUS='OLD')
      READ(11) X,XK,TGBD,NBODY,NBODYH,NBC,NWAKEL,NWAKEF
      READ(11) M1,M2,M3,M4
      IWL=2*NWAKEL-1
      IWF=2*NWAKEF
      READ(11) (ISEP(I),ZSSP(I),I=1,2)
      READ(11) (ZBODY(I),ZSBD(I),ALBD(I),I=1,NBODY)
      READ(11) ((AA(I,J),J=1,150),AX(I),I=1,150)
      READ(11) (GWAKE(I),ZWAKE(I),WCORE(I),I=1,IWL,2)
      READ(11) (GWAKE(I),ZWAKE(I),WCORE(I),I=2,IWF,2)
      CLOSE(UNIT=11)

      RETURN
      END

      SUBROUTINE INITIAL


      IMPLICIT COMPLEX Z

      COMMON/CORE/VAL1,VAL2
      COMMON/SEPRAT/ISEP(2),ZSSP(2)
      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,DELTA
      COMMON/TCSPLT/XMIN,YMIN,XMAX,YMAX
      COMMON/OPTION/INSTEP,NSTEP,NPLT,IFPL,IFFL,UMAX
      COMMON/MATRIX/NBC,AA(150,150),AX(150)
      COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)
      COMMON/PRESSURE/CPL(150),CPU(150),CPML(250),CPMU(150)
```

```
      DATA XMIN,XMAX,YMIN,YMAX/0.,1.2,-0.2,1./
      PI=3.1415927

C     INPUT PARAMETERS   ----------------------------------------
      ALPHD=22.
      EPSL=22.
      DELTAD=0.
      VAL1=2.5
      CAL2=1.5
      DX=0.02
      X0=1.
      A0=0.01
      NPLOT=250
      INSTEP=1
      NSTEP=500
      ALPH=ALPHD*PI/180.
      EPSL=EPSLD*PI/180.
      DELTA=DELTAD*PI/180.
      IFFL=
      IFPL=
      UMAX=
C     ------------------------------------------------------------

      IF(IFFL.NE.0) GO TO 11

C     DIFINE THE WALL POINTS  *  *  *
      CALL BODY
C

C     INITIAL STARTING CONDITIONS----------------------------------
      X=X0
      NWAKEL=0
      NWAKEF=0
      TGBD=0.
C     ------------------------------------------------------------

C     INITIALIZE OF MEAN PRESSURE  ---------------------------------
      DO 987 I=1,250
987   CPM(I)=0.
C     ------------------------------------------------------------


C     DEFINE GEOMETRY INFLUENCE MATRIX  *  *  *
      CALL GEOINF
C

C     GAUSS ELIMINATION--------------------------------------------
      NBC=NBODY+1
      DO 140 J=1,NBC-1
        AA(J,J)=1./AA(J,J)
        DO 140 I=J+1,NBC
        AA(I,J)=AA(I,J)*AA(J,J)
        DO 140 K=J+1,NBC
140   AA(I,K)=AA(I,K)-AA(I,J)*AA(J,K)
      AA(NBC,NBC)=1./AA(NBC,NBC)
C     ------------------------------------------------------------

11    RETURN
      END

C     ============================================================


      SUBROUTINE BODY
```

```
C   FORMATION OF BODY PANELS FOR THE FLAP PROBLEM   ------------

C       N1,N2 : NUMBER OF PANELS ON MAIN WING AND FLAP
C       XK    : RATIO OF MAIN WING AND TOTAL LENGTH

      IMPLICIT COMPLEX Z

      COMMON/BODY/M1,M2,M3,M4
      COMMON/SEPRAT/ISEP(2),ZSSP(2)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)
      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,DELTA

      DATA N1,N2/30, 25/
      DATA XK/0.618/

C   BODY COORDINATE FORMATION   ---------------------------------
      M1=N1+1
      M2=M1+N2
      M3=M2+1
      M4=M3+N1
      M5=M4+N2
      NBODY=M5
      NBODYH=(NBODY+1)/2
      ISEP(1)=M2
      ISEP(2)=M1

      ZBODY(M2)=CMPLX(0.,-XK)+(CMPLX(0.,-1.)-CMPLX(0.,-XK))
     *                        *CEXP(-CMPLX(0.,1.)*DELTA)
      ZBODY(M3)=CONJG(ZBODY(M2))
      DIS=XK/FLOAT(N1)
      ZDIS1=(ZBODY(M2)-CMPLX(0.,-XK))/FLOAT(N2)
      ZDIS2=(CMPLX(0.,XK)-ZBODY(M3))/FLOAT(N2)

CC   LOWER PANELS   ------------------------------
      DO 200 I=1,M1
        YY=-DIS*FLOAT(I-1)
200   ZBODY(I)=CMPLX(0.,YY)
      DO 300 I=M1+1,M2-1
300   ZBODY(I)=ZBODY(M1)+ZDIS1*FLOAT(I-M1)
CC   ------------------------------------------

CC   UPPER PANELS   ------------------------------
      DO 400 I=M3,M4
400   ZBODY(I)=ZBODY(M2)+ZDIS2*FLOAT(I-M3)
      DO 500 I=M4+1,M5
      YY=XK-DIS*FLOAT(I-M4)
500   ZBODY(I)=CMPLX(0.,YY)
CC   ------------------------------------------

CC  DEFINE PANEL SLOPE AND LENGTH----------------
      DO 300 K=1,NBODYH-1
        ZZ=ZBODY(1+K)-ZBODY(K)
        ALBD(K)=CABS(ZZ)
300   ZSBD(K)=ZZ/ALBD(K)
      DO 400 K=NBODY,NBODYH+1,-1
        KK=1+MOD(K+NBODY,NBODY)
        ZZ=ZBODY(K)-ZBODY(KK)
        ALBD(K-1)=CABS(ZZ)
400   ZSBD(K-1)=ZZ/ALBD(K-1)
CC   ------------------------------------------
C-------------------------------------------------------------------

C   AT SEPARATION POINTS   ---------------------------------------
      ZSSP(1)=ZSBD(M2)
      ZSSP(2)=ZSBD(M1)
```

```
C     ------------------------------------------------------------

      RETURN
      END

C=============================================================


      SUBROUTINE GEOINF

C     SUBROUTINE FOR CONSTRUCTING THE GEOMETRIC INFLUENCE COEFFICIENTS

      IMPLICIT COMPLEX Z

      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,DELTA
      COMMON/MATRIX/NBC,AA(150,150),AX(150)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)

C     FIRST ROW OF THE MATRIX WHICH REPRESENTS TOTAL BOUND VORTEX
C     STRENGTH EQUALS TOTAL WAKE STRENGTH    ----------------------

CC    LOWER PART    --------------------------------
      J=0
      JJ=NBODY-1
      AJM1=ALBD(JJ)
      DO 100 JJ=1,NBODYH-1
        J=J+1
        AJ=ALBD(JJ)
        AA(1,J)=0.5*(AJ+AJM1)
100   AJM1=AJ
      J=NBODYH
      AA(1,J)=0.5*AJM1
CC    ---------------------------------------------

CC    UPPER PART    --------------------------------
      J=NBODY+1
      JJ=NBODY-1
      AJP1=ALBD(JJ)
      DO 200 JJ=NBODY-1,NBODYH+1,-1
        J=J-1
        AJ=ALBD(JJ-1)
        AA(1,J)=0.5*(AJ+AJP1)
200   AJP1=AJ
      J=NBODYH+1
      AA(1,J)=0.5*AJP1
CC    ---------------------------------------------

      J=NBODY+1
      AA(1,J)=0.
C-------------------------------------------------------------

C     SECOND TO LAST ROWS-----------------------------------------
      I=1
      DO 300 II=1,NBODY
        I=I+1
        ZI=ZBODY(II)
        JJJ=NBODY-1
        AJ=ALBD(JJJ)

        IF(II.EQ.1) THEN
        ALOGJ=ALOG(AJ)
        TIJM=AJ*(0.5*ALOGJ-0.75)
        ELSE IF(II.EQ.NBODY) THEN
        ALOGJ=ALOG(AJ)
        TIJM=AJ*(ALOGJ*0.5-0.25)
```

```
        ELSE
        ZMID=0.5*(ZBODY(NBODY)+ZBODY(1))
        DIJH=ALOG(CABS(ZI-ZMID))
        DIJP=ALOG(CABS(ZI-ZBODY(NBODY)))
        TIJM=AJ/6.*(2.*DIJH+DIJP)
        ENDIF
        J=0

CC    LOWER PART----------------------------------
        DO 400 JJ=1,NBODYH-1
          J=J+1
          JP1=1+J
          AJ=ALBD(JJ)
          IF(JP1.EQ.II) THEN
          ALOGJ=ALOG(AJ)
          RIJ=AJ*(ALOGJ-1.)
          TIJ=0.5*AJ*(ALOGJ-1.5)
          ELSE IF(JJ.EQ.II) THEN
          ALOGJ=ALOG(AJ)
          RIJ=AJ*(ALOGJ-1.)
          TIJ=0.5*AJ*(ALOGJ-0.5)
          ELSE
          ZMID=0.5*(ZBODY(JJ)+ZBODY(JP1))
          DIJ =ALOG(CABS(ZI-ZBODY(JJ)))
          DIJP=ALOG(CABS(ZI-ZBODY(JP1)))
          DIJH=ALOG(CABS(ZI-ZMID))
          RIJ=AJ/6.*(DIJ+4.*DIJH+DIJP)
          TIJ=AJ/6.*(    2.*DIJH+DIJP)
          ENDIF
          AA(I,J)=0.5/PI*(RIJ-TIJ+TIJM)
400     TIJM=TIJ

        J=NBODYH
        JJ=NBODYH
        AJ=ALBD(JJ-1)
        ALOGJ=ALOG(AJ)
        IF(II.EQ.JJ) THEN
        AA(I,J)=0.5*0.5/PI*(ALOGJ-1.5)*AJ
        ELSE IF(II.EQ.JJ-1) THEN
        AA(I,J)=0.5*0.5/PI*(ALOGJ-0.5)*AJ
        ELSE
        ZMID=0.5*(ZBODY(NBODYH-1)+ZBODY(NBODYH))
        DIJ=ALOG(CABS(ZI-ZBODY(NBODYH)))
        DIJH=ALOG(CABS(ZI-ZMID))
        AA(I,J)=0.5*AJ*(2.*DIJH+DIJ)/(6.*PI)
        ENDIF
CC    ------------------------------------------

CC    UPPER PART----------------------------------
        JJJ=NBODY-1
        AJ=ALBD(JJJ)
        IF(II.EQ.1) THEN
        ALOGJ=ALOG(AJ)
        TIJM=0.5*AJ*(ALOGJ-0.5)
        ELSE IF(II.EQ.NBODY) THEN
        ALOGJ=ALOG(AJ)
        TIJM=0.5*AJ*(ALOGJ-1.5)
        ELSE
        ZMID=0.5*(ZBODY(NBODY)+ZBODY(1))
        DIJ=ALOG(CABS(ZI-ZBODY(NBODY)))
        DIJH=ALOG(CABS(ZI-ZMID))
        TIJM=AJ/6.*(2.*DIJH+DIJ)
        ENDIF
        J=NBODY+1
        DO 500 JJ=NBODY,NBODYH+2,-1
          J=J-1
```

```
            JM1=JJ-1
            JJJ=JJ-1
            AJ=ALBD(JJJ)
            IF(JM1.EQ.II) THEN
            ALOGJ=ALOG(AJ)
            RIJ=AJ*(ALOGJ-1.)
            TIJ=0.5*AJ*(ALOGJ-1.5)
            ELSE IF(JJ.EQ.II) THEN
            ALOGJ=ALOG(AJ)
            RIJ=AJ*(ALOGJ-1.)
            TIJ=0.5*AJ*(ALOGJ-0.5)
            ELSE
            ZMID=0.5*(ZBODY(JJ)+ZBODY(JM1))
            DIJ=ALOG(CABS(ZI-ZBODY(JJ)))
            DIJM=ALOG(CABS(ZI-ZBODY(JM1)))
            DIJH=ALOG(CABS(ZI-ZMID))
            RIJ=AJ/6.*(DIJ+4.*DIJH+DIJM)
            TIJ=AJ/6.*(2.*DIJH+DIJM)
            ENDIF
            AA(I,J)=0.5/PI*(RIJ-TIJ+TIJM)
500         TIJM=TIJ

            J=NBODYH+1
            JJ=NBODYH+1
            JP1=JJ+1
            JJJ=NBODYH+1
            AJ=ALBD(JJJ)
            IF(JJ.EQ.II) THEN
            AA(I,J)=0.5*0.5/PI*(ALOGJ-1.5)*AJ
            ELSE IF(II.EQ.JP1) THEN
            AA(I,J)=0.5*0.5/PI*(ALOGJ-0.5)*AJ
            ELSE
            ZMID=0.5*(ZBODY(JJ)+ZBODY(JP1))
            DIJ=ALOG(CABS(ZI-ZBODY(JJ)))
            DIJH=ALOG(CABS(ZI-ZMID))
            AA(I,J)=0.5*AJ*(2.*DIJH+DIJ)/(6.*PI)
            ENDIF
CC  ------------------------------------------------

300     CONTINUE
C -------------------------------------------------------------

C     LAST COLUMN-----------------------------------------------
          J=NBODY+1
          DO 600 II=1,NBODY
            I=II+1
600       AA(I,J)=1.
C  -------------------------------------------------------------

          RETURN
          END       SUBROUTINE SOLVE

C     THIS SUBROUTINE CACULATES THE STRENGTH OF BOUND VORTICES    --

          IMPLICIT COMPLEX Z

          COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,T,DELTA
          COMMON/MATRIX/NBC,AA(150,150),AX(150)
          COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)
          COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)

C     APPLY THE B.C. FOR THE WAKE INFLUENCE-----------------------
C     RIGHT HAND SIDE OF THE MATRICES-----------------------------
          I=1
          AX(I)=TGBD
          DO 100 II=1,NBODY
```

```
        I=I+1
        ZI=ZBODY(II)
        PSI=0.
        IWKL=NWAKEL*2-1
        IWKF=NWAKEF*2
        DO 120 J=1,IWKL,2
        ZC=CONJG(ZWAKE(J))
120   PSI=PSI-GWAKE(J)*ALOG(CABS(ZI-ZWAKE(J)))
     *                +GWAKE(J)*ALOG(CABS(ZI-ZC))
        DO 121 J=2,IWKF,2
          ZC=CONJG(ZWAKE(J))
121     PSI=PSI-GWAKE(J)*ALOG(CABS(ZI-ZWAKE(J)))
     *         +GWAKE(J)*ALOG(CABS(ZI-ZC))
100   AX(I)=0.5/PI*PSI-AIMAG(ZI)*SIN(ALPH)
C     ------------------------------------------------------------

C     SOLVE FOR THE VORTEX STRENGTH OF THE BOUNDED VORTICES--------
      DO 300 J=1,NBC-1
      DO 300 I=J+1,NBC
300   AX(I)=AX(I)-AX(J)*AA(I,J)
      DO 320 JJ=1,NBC-1
        J=NBC+1-JJ
        AX(J)=AX(J)*AA(J,J)
        DO 320 I=1,J-1
320   AX(I)=AX(I)-AX(J)*AA(I,J)
      AX(1)=AX(1)*AA(1,1)
C     ------------------------------------------------------------

      RETURN
      END
      SUBROUTINE MOVE(ISTEP)

C     NEW POSITION OF VORTICES AND NEW VORTEX GENERATION    --------

      IMPLICIT COMPLEX Z

      COMMON/WAKVEL/ZVEL(400)
      COMMON/SEPRAT/ISEP(2),ZSSP(2)
      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,T,DELTA
      COMMON/TCSPLT/XMIN,YMIN,XMAX,YMAX
      COMMON/OPTION/INSTEP,NSTEP,NPLT,IFPL,IFFL,UMAX
      COMMON/MATRIX/NBC,AA(150,150),AX(150)
      COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)

C     LIMIT THE BOUND VORTICITY STRENGTH NEAR THE TIPS-------------
      DO 11 I=ISEP(1)-3,ISEP(1)+3
11    IF(ABS(AX(I)).GT.UMAX) AX(I)=SIGN(UMAX,AX(I))
C     ------------------------------------------------------------

C     PUT SYMMETRIC CONDITIONS-------------------------------------
      DO 12 I=2,NBODYH
12    AX(NBODY+2-I)=-AX(I)
C     ------------------------------------------------------------

C     FOR THE VORTEX WAKE------------------------------------------
C     PUT NEW VORTEX PANNELS AT THE SEPARATION POINTS--------------
CC    FOR LEADING EDGE VORTEX SYSTEM    ------------
      JK=ISEP(1)
      CALL CRSVEL(2,JK,VL,VU,WL,WU)
      VM=(VL+VU)/2.
      WM=(WL+WU)/2.
      VSEP=SQRT(VM**2+WM**2)
      NWAKEL=NWAKEL+1
      IWKL=2*NWAKEL-1
      ASIGN=SIGN(1.,AX(ISEP(1)))
```

```
      GWAKE(IWKL)=0.5*ASIGN*DX*ABS(AX(ISEP(1)))**2/(X*TAN(EPSL))
      ZWAKE(IWKL)=ZBODY(ISEP(1))+DX*(ZSSP(1)*VSEP/2.)
     *                                /(X*TAN(EPSL))
      WCORE(IWKL)=A0
CC    ------------------------------------------

CC    FOR HINGE VORTEX SYSTEM   ------------------
      IF(ISTEP.LE.700) GO TO 1000
      IF(DELTA.EQ.0.) GO TO 1000
      ASIGN=SIGN(1.,AX(ISEP(2)))
      IF(NWAKEF.GE.1) GO TO 280
      JK=ISEP(2)
      CALL CRSVEL(?,JK,VL,VU,WL,WU)
      VM=(VU+VL)/2.
      WM=(WU+WL)/2.
      VSEP=SQRT(VM**2+WM**2)
      IF(WM.LT.0.) GO TO 934
      ZZ=ZBODY(ISEP(2))-ZBODY(ISEP(2)+1)
      ZSSP(2)=ZZ/CABS(ZZ)
      GO TO 280
934   ZZ=ZBODY(ISEP(2))-ZBODY(ISEP(2)-1)
      ZSSP(2)=ZZ/CABS(ZZ)
280   NWAKEF=NWAKEF+1
      IWKF=NWAKEF*2
      GWAKE(IWKF)=0.5*ASIGN*DX*ABS(AX(ISEP(2)))**2/(X*TAN(EPSL))
      ZWAKE(IWKF)=ZBODY(ISEP(2))+DX*(ZSSP(2)*U(ISEP(2))/2.)
     *                           /(X*TAN(EPSL))
      WCORE(IWKF)=A0
CC    ------------------------------------------
C     ------------------------------------------------------------

C     UPDATE THE VORTEX POSITIONS USING RK-4TH ORDER METHODS  *  *  *
1000  CALL RUKU4
C

      WRITE(6,*) ISTEP,NWAKEL,NWAKEF,GWAKE(1),ZWAKE(1),GWAKE(2),ZWAKE(2)
      WRITE(6,*) AX(ISEP(1)),AX(ISEP(2))

C     UPDATE THE VORTEX STRENGTH AND VORTEX POSITIONS--------------
      SLNGTH=1.+DX/X
      DO 210 II=1,NWAKEL
        I=II*2-1
        ZWAKE(I)=ZWAKE(I)/SLNGTH
        GWAKE(I)=GWAKE(I)/SLNGTH
        WCORE(I)=WCORE(I)/SQRT(SLNGTH)
210   CONTINUE

      DO 211 II=1,NWAKEF
        I=II*2
        ZWAKE(I)=ZWAKE(I)/SLNGTH
        GWAKE(I)=GWAKE(I)/SLNGTH
        WCORE(I)=WCORE(I)/SQRT(SLNGTH)
211   CONTINUE

      X=X+DX
C-------------------------------------------------------------

C     CHECK THE SURFACE MERGE  *  *  *
      CALL SMERGE(NNWL,NNWF)
      NWAKEL=NNWL
      NWAKEF=NNWF
C

C     CORE MERGING  *  *  *
      CALL CORE(NWL,NWF)
      NWAKEL=NWL
```

```
      NWAKEF=NWF
C
C   VORTEX MERGING  *  *  *
      CALL MERGE(ISTEP,NWL,NWF)
      NWAKEL=NWL
      NWAKEF=NWF
C
      RETURN
      END


C   ================================================================


      SUBROUTINE RUKU4

C   SCHEME FOR THE FOURTH ORDER OF RUNGE-KUTTA METHODS*************

      IMPLICIT COMPLEX Z

      COMMON/WALVEL/ZVL(150)
      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,T,DELTA
      COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)
      COMMON/RK4/ZWP(150),ZWPP(150),ZWPPP(150)

      DIMENSION ZVLP(150),ZVLPP(150),ZVLPPP(150)

C   FIRST STEP-----------------------------------------------------
      DO 100 J=1,NWAKEL
        II=J*2-1
        ZVL(II)=ZV(ZWAKE(II))
100   ZWP(II)=ZWAKE(II)+ZVL(II)*DX/(2.*X*TAN(EPSL))

      DO 101 J=1,NWAKEF
        II=2*J
        ZVL(II)=ZV(ZWAKE(II))
101   ZWP(II)=ZWAKE(II)+ZVL(II)*DX/(2.*X*TAN(EPSL))
C   -------------------------------------------------------------

C   SECOND STEP----------------------------------------------------
      DO 200 J=1,NWAKEL
        II=2*J-1
        ZVLP(II)=ZVP(ZWP(II))
200   ZWPP(II)=ZWAKE(II)+ZVLP(II)*DX/(2.*X*TAN(EPSL))

      DO 201 J=1,NWAKEF
        II=2*J
        ZVLP(II)=ZVP(ZWP(II))
201   ZWPP(II)=ZWAKE(II)+ZVLP(II)*DX/(2.*X*TAN(EPSL))
C   -------------------------------------------------------------

C  THIRD STEP-----------------------------------------------------
      DO 300 J=1,NWAKEL
        II=2*J-1
        ZVLPP(II)=ZVPP(ZWPP(II))
300   ZWPPP(II)=ZWAKE(II)+ZVLPP(II)*DX/(X*TAN(EPSL))

      DO 301 J=1,NWAKEF
        II=2*J
        ZVLPP(II)=ZVPP(ZWPP(II))
301   ZWPPP(II)=ZWAKE(II)+ZVLPP(II)*DX/(X*TAN(EPSL))
C   -------------------------------------------------------------
```

```
C   FINAL STEP-----------------------------------------------
      DO 400 J=1,NWAKEL
      II=2*J-1
400   ZVLPPP(II)=ZVPPP(ZWPPP(II))

      DO 401 J=1,NWAKEF
      II=2*J
401   ZVLPPP(II)=ZVPPP(ZWPPP(II))

      DO 500 J=1,NWAKEL
      II=2*J-1
500   ZWAKE(II)=ZWAKE(II)+(ZVL(II)+2.*ZVLP(II)
     *                 +2.*ZVLPP(II)+ZVLPPP(II))*DX/(6.*X*TAN(EPSL))

      DO 501 J=1,NWAKEF
      II=2*J
501   ZWAKE(II)=ZWAKE(II)+(ZVL(II)+2.*ZVLP(II)
     *               +2.*ZVLPP(II)+ZVLPPP(II))*DX/(6.*X*TAN(EPSL))
C   ---------------------------------------------------------

      RETURN
      END

C   =========================================================



      COMPLEX FUNCTION ZV(ZI)

C   CALCULATION OF INDUCED VELOCITY BY BOUND AND WAKE VORTICES

      PARAMETER ZIMG=CMPLX(0.,1.)

      IMPLICIT COMPLEX (Z,V)

      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,DELTA
      COMMON/MATRIX/NBC,AA(150,150),AX(150)
      COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)

      VI=0.

C   INDUCE VELOCITY BY THE BOUNDED VORTEX PANELS---------------
      F00=GFUN(ZI,ZBODY(1),A0)
      DO 100 I=1,NBODYH-1
        IP1=I+1
        Z12=0.5*(ZBODY(I)+ZBODY(IP1))
        F12=GFUN(ZI,Z12,A0)
        FP1=GFUN(ZI,ZBODY(IP1),A0)
        AI =ALBD(I)
        H0=F00+4.*F12+FP1
        H1=(   2.*F12+FP1)*AI
        H2=(      F12+FP1)*AI**2
        Z00=ZI-ZBODY(I)
        ZP=ZIMG*(-H0*Z00+H1*ZSBD(I))*AI/6.
        ZQ=ZIMG*(-H1*Z00+H2*ZSBD(I))/6.
        VI=VI+AX(I)*ZP+(AX(IP1)-AX(I))*ZQ
100   F00=FP1

      F00=GFUN(ZI,ZBODY(1),A0)
      DO 200 I=NBODY,NBODYH+1,-1
        IP1=1+MOD(I+NBODY,NBODY)
        Z12=0.5*(ZBODY(I)+ZBODY(IP1))
        F12=GFUN(ZI,Z12,A0)
        FP1=GFUN(ZI,ZBODY(I),A0)
```

```
        AI=ALBD(I-1)
        H0=F00+4.*F12+FP1
        H1=(2.*F12+FP1)*AI
        H2=(F12+FP1)*AI**2
        Z00=ZI-ZBODY(IP1)
        ZP=ZIMG*(-H0*Z00+H1*ZSBD(I-1))*AI/6.
        ZQ=ZIMG*(-H1*Z00+H2*ZSBD(I-1))/6.
        VI=VI+AX(IP1)*ZP+(AX(I)-AX(IP1))*ZQ
200   F00=FP1
C     ------------------------------------------------------------

C   INDUCED VELOCITY BY THE FREE VORTEX-------------------------
      DO 300 J=1,NWAKEL
        II=2*J-1
        R=WCORE(II)
        ZC=CONJG(ZWAKE(II))
        VI=VI-GWAKE(II)*GFUN(ZI,ZWAKE(II),R)*ZIMG*(ZI-ZWAKE(II))
     *         +GWAKE(II)*GFUN(ZI,ZC,R)*ZIMG*(ZI-ZC)
300   CONTINUE

      DO 301 J=1,NWAKEF
        II=2*J
        R=WCORE(II)
        ZC=CONJG(ZWAKE(II))
        VI=VI-GWAKE(II)*GFUN(ZI,ZWAKE(II),R)*ZIMG*(ZI-ZWAKE(II))
     *         +GWAKE(II)*GFUN(ZI,ZC,R)*ZIMG*(ZI-ZC)
301   CONTINUE
C     ------------------------------------------------------------

C     ADD THE FREE STREAM VELOCITY--------------------------------
      ZV=0.5/PI*VI+CMPLX(1.,0.)*SIN(ALPH)
C     ------------------------------------------------------------

      RETURN
      END

C     ============================================================


      FUNCTION GFUN(ZI,ZJ,R)

C     CALCULATION OF INDECED VELOCITY BETWEEN TWO VORTICES    ------

      IMPLICIT COMPLEX Z

      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,T,DELTA

      AL=CABS(ZI-ZJ)
      IF(AL.LT.R) THEN
      X=(AL/R)**2
      GFUN=3./R**2*(1.+X*(X/3.-1.))
      ELSE
      GFUN=1./AL**2
      ENDIF

      RETURN
      END

C     ============================================================

      COMPLEX FUNCTION ZVP(ZPI)

C     CALCULATION OF VELOCITY BY BOUND AND WAKE VORTICES    --------

      PARAMETER ZIMG=CMPLX(0.,1.)
```

```
      IMPLICIT COMPLEX (Z,V)

      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,T,DELTA
      COMMON/MATRIX/NBC,AA(150,150),AX(150)
      COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)
      COMMON/RK4/ZWP(150),ZWPP(150),ZWPPP(150)

      VPI=0.

C   INDUCE VELOCITY BY THE BOUNDED VORTEX PANELS----------------
      F00=GFUN(ZPI,ZBODY(1),A0)
      DO 100 I=1,NBODYH-1
        IP1=1+MOD(I+NBODY,NBODY)
        Z12=0.5*(ZBODY(I)+ZBODY(IP1))
        F12=GFUN(ZPI,Z12,A0)
        FP1=GFUN(ZPI,ZBODY(IP1),A0)
        AI =ALBD(I)
        H0=F00+4.*F12+FP1
        H1=(   2.*F12+FP1)*AI
        H2=(      F12+FP1)*AI**2
        Z00=ZPI-ZBODY(I)
        ZP=ZIMG*(-H0*Z00+H1*ZSBD(I))*AI/6.
        ZQ=ZIMG*(-H1*Z00+H2*ZSBD(I))/6.
        VPI=VPI+AX(I)*ZP+(AX(IP1)-AX(I))*ZQ
100   F00=FP1

      F00=GFUN(ZPI,ZBODY(1),A0)
      DO 200 I=NBODY,NBODYH+1,-1
        IP1=1+MOD(I+NBODY,NBODY)
        Z12=0.5*(ZBODY(I)+ZBODY(IP1))
        F12=GFUN(ZPI,Z12,A0)
        FP1=GFUN(ZPI,ZBODY(I),A0)
        AI=ALBD(I-1)
        H0=F00+4.*F12+FP1
        H1=(2.*F12+FP1)*AI
        H2=(F12+FP1)*AI**2
        Z00=ZPI-ZBODY(IP1)
        ZP=ZIMG*(-H0*Z00+H1*ZSBD(I-1))*AI/6.
        ZQ=ZIMG*(-H1*Z00+H2*ZSBD(I-1))/6.
        VPI=VPI+AX(IP1)*ZP+(AX(I)-AX(IP1))*ZQ
200   F00=FP1
C   ------------------------------------------------------------

C  INDUCED VELOCITY BY THE FREE VORTEX--------------------------
      DO 300 J=1,NWAKEL
        II=J*2-1
        R=WCORE(II)
        ZC=CONJG(ZWP(II))
        VPI=VPI-GWAKE(II)*GFUN(ZPI,ZWP(II),R)*ZIMG*(ZPI-ZWP(II))
     *          +GWAKE(II)*GFUN(ZPI,ZC,R)*ZIMG*(ZPI-ZC)
300   CONTINUE

      DO 301 J=1,NWAKEF
        II=J*2
        R=WCORE(II)
        ZC=CONJG(ZWP(II))
        VPI=VPI-GWAKE(II)*GFUN(ZPI,ZWP(II),R)*ZIMG*(ZPI-ZWP(II))
     *          +GWAKE(II)*GFUN(ZPI,ZC,R)*ZIMG*(ZPI-ZC)
301   CONTINUE
C   ------------------------------------------------------------

C   ADD THE FREE STREAM VELOCITY---------------------------
      ZVP=0.5/PI*VPI+CMPLX(1.,0.)*SIN(ALPH)
C   ------------------------------------------------------
```

```
      RETURN
      END
C     ============================================================
      COMPLEX FUNCTION ZVPP(ZPPI)
C     THIS SUBROUTINE CACULATE THE VELOCITY DUE TO BODY PANNEL    --
      PARAMETER ZIMG=CMPLX(0.,1.)
      IMPLICIT COMPLEX (Z,V)
      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,T,DELTA
      COMMON/MATRIX/NBC,AA(150,150),AX(150)
      COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)
      COMMON/RK4/ZWP(150),ZWPP(150),ZWPPP(150)
      VPPI=0.
C     INDUCE VELOCITY BY THE BOUNDED VORTEX PANELS-----------------
      F00=GFUN(ZPPI,ZBODY(1),A0)
      DO 100 I=1,NBODYH-1
        IP1=1+MOD(I+NBODY,NBODY)
        Z12=0.5*(ZBODY(I)+ZBODY(IP1))
        F12=GFUN(ZPPI,Z12,A0)
        FP1=GFUN(ZPPI,ZBODY(IP1),A0)
        AI =ALBD(I)
        H0=F00+4.*F12+FP1
        H1=(   2.*F12+FP1)*AI
        H2=(      F12+FP1)*AI**2
        Z00=ZPPI-ZBODY(I)
        ZP=ZIMG*(-H0*Z00+H1*ZSBD(I))*AI/6.
        ZQ=ZIMG*(-H1*Z00+H2*ZSBD(I))/6.
        VPPI=VPPI+AX(I)*ZP+(AX(IP1)-AX(I))*ZQ
100   F00=FP1
      F00=GFUN(ZPPI,ZBODY(1),A0)
      DO 200 I=NBODY,NBODYH+1,-1
        IP1=1+MOD(I+NBODY,NBODY)
        Z12=0.5*(ZBODY(I)+ZBODY(IP1))
        F12=GFUN(ZPPI,Z12,A0)
        FP1=GFUN(ZPPI,ZBODY(I),A0)
        AI=ALBD(I-1)
        H0=F00+4.*F12+FP1
        H1=(2.*F12+FP1)*AI
        H2=(F12+FP1)*AI**2
        Z00=ZPPI-ZBODY(IP1)
        ZP=ZIMG*(-H0*Z00+H1*ZSBD(I-1))*AI/6.
        ZQ=ZIMG*(-H1*Z00+H2*ZSBD(I-1))/6.
        VPPI=VPPI+AX(IP1)*ZP+(AX(I)-AX(IP1))*ZQ
200   F00=FP1
C     ------------------------------------------------------------
C   INDUCED VELOCITY BY THE FREE VORTEX----------------------
      DO 300 J=1,NWAKEL
        II=J*2-1
        R=WCORE(II)
        ZC=CONJG(ZWPP(II))
        VPPI=VPPI-GWAKE(II)*GFUN(ZPPI,ZWPP(II),R)
     *                              *ZIMG*(ZPPI-ZWPP(II))
     *           +GWAKE(II)*GFUN(ZPPI,ZC,R)*ZIMG*(ZPPI-ZC)
300   CONTINUE
      DO 301 J=1,NWAKEF
```

```
          II=J*2
          R=WCORE(II)
          ZC=CONJG(ZWPP(II))
          VPPI=VPPI-GWAKE(II)*GFUN(ZPPI,ZWPP(II),R)
     *                              *ZIMG*(ZPPI-ZWPP(II))
     *  . .          +GWAKE(II)*GFUN(ZPPI,ZC,R)*ZIMG*(ZPPI-ZC)
301     CONTINUE
C     ------------------------------------------------------------

C     ADD THE FREE STREAM VELOCITY----------------------------
        ZVPP=0.5/PI*VPPI+CMPLX(1.,0.)*SIN(ALPH)
C     ------------------------------------------------------------

        RETURN
        END

C     ================================================================



        COMPLEX FUNCTION ZVPPP(ZPPPI)

C     THIS SUBROUTINE CACULATE THE VELOCITY DUE TO BODY PANNEL    --

        PARAMETER ZIMG=CMPLX(0.,1.)

        IMPLICIT COMPLEX (Z,V)

        COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,T,DELTA
        COMMON/MATRIX/NBC,AA(150,150),AX(150)
        COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)
        COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)
        COMMON/RK4/ZWP(150),ZWPP(150),ZWPPP(150)

        VPPPI=0.

C     INDUCE VELOCITY BY THE BOUNDED VORTEX PANELS-----------------
        F00=GFUN(ZPPPI,ZBODY(1),A0)
        DO 100 I=1,NBODYH-1
          IP1=1+MOD(I+NBODY,NBODY)
          Z12=0.5*(ZBODY(I)+ZBODY(IP1))
          F12=GFUN(ZPPPI,Z12,A0)
          FP1=GFUN(ZPPPI,ZBODY(IP1),A0)
          AI =ALBD(I)
          H0=F00+4.*F12+FP1
          H1=(    2.*F12+FP1)*AI
          H2=(       F12+FP1)*AI**2
          Z00=ZPPPI-ZBODY(I)
          ZP=ZIMG*(-H0*Z00+H1*ZSBD(I))*AI/6.
          ZQ=ZIMG*(-H1*Z00+H2*ZSBD(I))/6.
          VPPPI=VPPPI+AX(I)*ZP+(AX(IP1)-AX(I))*ZQ
100     F00=FP1

        F00=GFUN(ZPPPI,ZBODY(1),A0)
        DO 200 I=NBODY,NBODYH+1,-1
          IP1=1+MOD(I+NBODY,NBODY)
          Z12=0.5*(ZBODY(I)+ZBODY(IP1))
          F12=GFUN(ZPPPI,Z12,A0)
          FP1=GFUN(ZPPPI,ZBODY(I),A0)
          AI=ALBD(I-1)
          H0=F00+4.*F12+FP1
          H1=(2.*F12+FP1)*AI
          H2=(F12+FP1)*AI**2
          Z00=ZPPPI-ZBODY(IP1)
          ZP=ZIMG*(-H0*Z00+H1*ZSBD(I-1))*AI/6.
          ZQ=ZIMG*(-H1*Z00+H2*ZSBD(I-1))/6.
```

```
        VPPPI=VPPPI+AX(IP1)*ZP+(AX(I)-AX(IP1))*ZQ
200   F00=FP1
C   ---------------------------------------------------------------

C  INDUCED VELOCITY BY THE FREE VORTEX--------------------------
      DO 300 J=1,NWAKEL
        II=J*2-1
        R=WCORE(II)
        ZC=CONJG(ZWPPP(II))
        VPPPI=VPPPI-GWAKE(II)*GFUN(ZPPPI,ZWPPP(II),R)*ZIMG*
     *                              (ZPPPI-ZWPPP(II))
     *              +GWAKE(II)*GFUN(ZPPPI,ZC,R)*ZIMG*(ZPPPI-ZC)
300   CONTINUE

      DO 301 J=1,NWAKEF
        II=J*2
        R=WCORE(II)
        ZC=CONJG(ZWPPP(II))
        VPPPI=VPPPI-GWAKE(II)*GFUN(ZPPPI,ZWPPP(II),R)*ZIMG*
     *                             (ZPPPI-ZWPPP(II))
     *              +GWAKE(II)*GFUN(ZPPPI,ZC,R)*ZIMG*(ZPPPI-ZC)
301   CONTINUE
C   ---------------------------------------------------------------

C   ADD THE FREE STREAM VELOCITY---------------------------------
      ZVPPP=0.5/PI*VPPPI+CMPLX(1.,0.)*SIN(ALPH)
C   ---------------------------------------------------------------

      RETURN
      END


C   ===============================================================

      SUBROUTINE SMERGE(NWL,NWF)

C   SURFACE MERGING SCHEME   -------------------------------------

      IMPLICIT COMPLEX Z

      COMMON/SEPRAT/ISEP(2),ZSSP(2)
      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,DELTA
      COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)

      NWL=NWAKEL
      NWF=NWAKEF

      IF(NWL.LE.0.) GO TO 501
      J=1

C   CHECK THE DISTANCE BETWEEN THE LEADING EDGE VORTICES AND THE SURFACE
500   YY=AIMAG(ZWAKE(J))
      XX=REAL(ZWAKE(J))
      IF(ABS(YY).GT.1.) GO TO 100
      IF(ABS(YY).LE.XK) GO TO 10
      RS=(AIMAG(ZBODY(ISEP(2)))-AIMAG(ZBODY(ISEP(1))))/
     1          (REAL(ZBODY(ISEP(2)))+0.005-REAL(ZBODY(ISEP(1))))
     2        *(XX-REAL(ZBODY(ISEP(1))))+AIMAG(ZBODY(ISEP(1)))
      IF(YY.LE.RS) GO TO 100
      GO TO 20
10    IF (XX.GT.0.005) GO TO 100
C   ---------------------------------------------------------------

C   READJUST THE VORTEX INDICES----------------------------------
20    IWL=NWL*2-1
```

```
      DO 300 K=J+2,IWL,2
        KK=K-2
        ZWAKE(KK)=ZWAKE(K)
        GWAKE(KK)=GWAKE(K)
        WCORE(KK)=WCORE(K)
300   CONTINUE
      NWL=NWL-1
      GO TO 30
C     ------------------------------------------------------------

100   J=J+2
30    JK=NWL*2-1
      IF(J.GT.JK) GO TO 400
      GO TO 500

400      J=2

C     CHECK THE DISTANCE BETWEEN THE HINGE VORTICES AND THE SURFACE
501   IF(NWF.LE.2) GO TO 401

999   YY=AIMAG(ZWAKE(J))
      XX=REAL(ZWAKE(J))
      IF(ABSYY).GT.1.) GO TO 101
      IF(ABS(YY).LE.XK) GO TO 11
      RS=(AIMAG(ZBODY(ISEP(2)))-AIMAG(ZBODY(ISEP(1))))/
     1          (REAL(ZBODY(ISEP(2)))+0.005-REAL(ZBODY(ISEP(1))))
     2            *(XX-REAL(ZBODY(ISEP(1))))+AIMAG(ZBODY(ISEP(1)))
      IF(YY.LE.RS) GO TO 101
      GO TO 21
11    IF (XX.GT.0.005) GO TO 101
C     ------------------------------------------------------------

C     READJUST THE VORTEX INDICES---------------------------------
21    IWF=NWF*2
      DO 301 K=J+2,IWF,2
        KK=K-2
        ZWAKE(KK)=ZWAKE(K)
        GWAKE(KK)=GWAKE(K)
        WCORE(KK)=WCORE(K)
301   CONTINUE
      NWF=NWF-1
      GO TO 31
C     ------------------------------------------------------------

101   J=J+2
31    JK=NWF*2-4
      IF(J.GE.JK) GO TO 401
      GO TO 999

401   RETURN
      END



C     ============================================================


      SUBROUTINE MERGE(NWL,NWF)

C     SCHEME WHEN TWO VORITICES ARE TOO CLOSE THE THOSE ARE MERGED TO ONE

      IMPLICIT COMPLEX Z

      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,T,DELTA
      COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(400),GWAKE(400),WCORE(400)
```

```
      NWL=NWAKEL
      NWF=NWAKEF

      IWL=NWL*2-1
      IWF=NWF*2

      IF(NWL.LT.5) GO TO 700

C     FOR LEADING EDGE VORTEX SYSTEM    ----------------------------

      J=1

C     CHECK THE DISTANCE BETWEEN TWO VORTICES----------------------
100   K=1
200   IF(J.EQ.K) GO TO 600
      DIS=CABS(ZWAKE(J)-ZWAKE(K))
      IF(DIS.LT.A0) GO TO 300
      GO TO 600
C     ------------------------------------------------------------

C     CLOSER THAN THE INITIAL CORE RADIUS THAN MERGED TO ONE-------
300   ZWJ=(ZWAKE(J)*GWAKE(J)+ZWAKE(K)*GWAKE(K))/(GWAKE(J)+GWAKE(K))
      GWJ=GWAKE(J)+GWAKE(K)
      WCJ=ABS((GWAKE(J)*WCORE(J)**3+GWAKE(K)*WCORE(K)**3)/GWJ)**(1./3.)
      ZWAKE(J)=ZWJ
      GWAKE(J)=GWJ
      WCORE(J)=WCJ
C     ------------------------------------------------------------

C     REARRANGE THE INDICES----------------------------------------
      DO 400 JN=K,IWL-2,2
        GWAKE(JN)=GWAKE(JN+2)
        ZWAKE(JN)=ZWAKE(JN+2)
        WCORE(JN)=WCORE(JN+2)
400   CONTINUE
      NWL=NWL-1
      IWL=NWL*2-1
600   IF(K.GE.IWL) GO TO 500
      K=K+2
      GO TO 200

500   IF(NWF.LT.3) GO TO 3
      KK=2
4     DIS=CABS(ZWAKE(J)-ZWAKE(KK))
      IF(DIS.LT.A0) GO TO 1
      GO TO 2
1     GWJ=GWAKE(J)+GWAKE(KK)
      IF(GWJ.LT.0.) GO TO 6
      ZWJ=(ZWAKE(J)*GWAKE(J)+ZWAKE(KK)*GWAKE(KK))/(GWAKE(J)+GWAKE(KK))
      WCJ=ABS((GWAKE(J)*WCORE(J)**3+
     *            GWAKE(KK)*WCORE(KK)**3)/GWJ)**(1./3.)
      ZWAKE(J)=ZWJ
      GWAKE(J)=GWJ
      WCORE(J)=WCJ
C     ------------------------------------------------------------

C     REARRANGE THE INDICES----------------------------------------
      DO 5 JN=KK,IWF-2,2
        GWAKE(JN)=GWAKE(JN+2)
        ZWAKE(JN)=ZWAKE(JN+2)
        WCORE(JN)=WCORE(JN+2)
5     CONTINUE
      NWF=NWF-1
      IWF=NWF*2
2     IF(KK.GE.IWF) GO TO 3
      KK=KK+2
```

```
      GO TO 4

6     ZWJ=(ZWAKE(J)*GWAKE(J)+ZWAKE(KK)*GWAKE(KK))/(GWAKE(J)+GWAKE(KK))
      WCJ=ABS((GWAKE(J)*WCORE(J)**3+
     *          GWAKE(KK)*WCORE(KK)**3)/GWJ)**(1./3.)
      ZWAKE(KK)=ZWJ
      GWAKE(KK)=GWJ
      WCORE(KK)=WCJ
C     ------------------------------------------------------------

C     REARRANGE THE INDICES-------------------------------------
      DO 11 JN=J,IWL-2,2
        GWAKE(JN)=GWAKE(JN+2)
        ZWAKE(JN)=ZWAKE(JN+2)
        WCORE(JN)=WCORE(JN+2)
11    CONTINUE
      NWL=NWL-1
      IWL=NWL*2-1

3     IF(J.GE.IWL) GO TO 700
      J=J+2
      GO TO 100
C     ------------------------------------------------------

C     FOR THE FLAP HIGE VORTEX SYSTEM ------------------------
C700   IF(ISTEP.LT.15) GO TO 906

700   IF(NWF.LT.3) GO TO 906

      J=2
900   K=2
901   IF(J.EQ.K) GO TO 902
      DIS=CABS(ZWAKE(J)-ZWAKE(K))
      IF(DIS.LT.A0) GO TO 903
      GO TO 902
903   ZWJ=(ZWAKE(J)*GWAKE(J)+ZWAKE(K)*GWAKE(K))/(GWAKE(K)+GWAKE(J))
      GWJ=GWAKE(J)+GWAKE(K)
      WCJ=ABS((GWAKE(J)*WCORE(J)**3+GWAKE(K)*WCORE(K)**3)/GWJ)**(1./3.)
      ZWAKE(J)=ZWJ
      GWAKE(J)=GWJ
      WCORE(J)=WCJ
      DO 904 JN=K,IWF-2,2
        GWAKE(JN)=GWAKE(JN+2)
        ZWAKE(JN)=ZWAKE(JN+2)
        WCORE(JN)=WCORE(JN+2)
904   CONTINUE
      NWF=NWF-1
      IWF=NWF*2
902   IF(K.GE.IWF) GO TO 905
      K=K+2
      GO TO 901
905   IF(J.GE.IWF) GO TO 906
      J=J+2
      GO TO 900
C     ------------------------------------------------------

906   RETURN
      END


C     ============================================================



      SUBROUTINE CORE(NWL,NWF)
```

```
C   MODEL FOR THE CORE REGION   --------------------------------

      IMPLICIT COMPLEX Z

      COMMON/CORE/VAL1,VAL2
      COMMON/SEPRAT/ISEP(2),ZSSP(2)
      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,T,DELTA
      COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)

      NWL=NWAKEL
      NWF=NWAKEF

      IWL=NWL*2-1
      IWF=NWF*2

      IF(NWL.LT.5) GO TO 600

C   FOR LEADING EDGE VORTEX SYSTEM  ---------------------------

      BTA=0.
      ZZ=ZBODY(ISEP(1))
      Z0=ZWAKE(1)
      ZARCM=CLOG(ZZ-Z0)
      ARCM=AIMAG(ZARCM)
      N=0

C   CALCULATE THE INCREMENT OF ROTATION ANGLE-----------------
      DO 100 J=IWL,3,-2
      II=J
      N=N+1
      ZJ=ZWAKE(J)
      ZARC=CLOG(ZJ-Z0)
      ARC=AIMAG(ZARC)
      ARCI=ARC-ARCM

C   HERE ASSUMED THAT THE ROTATION CANNOT EXCEED PI BETWEEN NEIGHBORING
C   VORTICES--------------------------------------------------

      IF(ABS(ARCI).LT.PI) GO TO 101
      SGN=SIGN(1.,ARCI)
      ARCI=-1.*SGN*ABS(ABS(ARCI)-2.*PI)
101   BTA=BTA+ARCI
C   ------------------------------------------------

C   SHEET ROTATION IS LARGER THAN 2PI, THAN IT MERGE TO CORE-----
      VAL=VAL1*PI
      IF(ABS(BTA).GT.ABS(VAL)) GO TO 102
      IF(J.EQ.3) GO TO 500
      ZZ=ZJ
100   ARCM=ARC
102   NWL=N
      ZCEN1=0.
      GCORE1=0.
      CORE1=0.
      DO 103 I=II,1,-2
      GCORE1=GCORE1+GWAKE(I)
      ZCEN1=ZCEN1+GWAKE(I)*ZWAKE(I)
      CORE1=CORE1+GWAKE(I)*WCORE(I)**3
103   CONTINUE
      GWAKE(1)=GCORE1
      ZWAKE(1)=ZCEN1/GCORE1
      WCORE(1)=(CORE1/GCORE1)**(1./3.)
C   ------------------------------------------------------------

C   RERRANGE THE INDICES OF THE VORTICES--------------------
```

```
      NI=(NWAKEL-NWL)*2
      IWL=NWL*2-1
      DO 104 JJ=3,IWL,2
      ZC1=ZWAKE(JJ+NI)
      GC1=GWAKE(JJ+NI)
      WC1=WCORE(JJ+NI)
      ZWAKE(JJ)=ZC1
      GWAKE(JJ)=GC1
104   WCORE(JJ)=WC1
      GO TO 600
500   NWL=NWAKEL
C     ------------------------------------------------------------
C     FOR FLAP HINGE VORTEX SYSTEM  --------------------------

600    IF(NWF.LT.3) GO TO 906

      BTA=0.
      ZZ=ZBODY(ISEP(2))
      Z0=ZWAKE(2)
      ZARCM=CLOG(ZZ-Z0)
      ARCM=AIMAG(ZARCM)
      N=0

C     CALCULATE THE INCREMENT OF ROTATION ANGLE-----------------
      DO 900 J=IWF,4,-2
      II=J
      N=N+1
      ZJ=ZWAKE(J)
      ZARC=CLOG(ZJ-Z0)
      ARC=AIMAG(ZARC)
      ARCI=ARC-ARCM

C     HERE ASSUMED THAT THE ROTATION CANNOT EXCEED PI BETWEEN NEIGHBORING
C     VORTICES-------------------------------------------------

      IF(ABS(ARCI).LT.PI) GO TO 901
      SGN=SIGN(1.,ARCI)
      ARCI=-1.*SGN*ABS(ABS(ARCI)-2.*PI)
901   BTA=BTA+ARCI
C     ------------------------------------------------------

C     SHEET ROTATION IS LARGER THAN 2PI, THAN IT MERGE TO CORE-----
      VAL=-SN*VAL2*PI
      IF(ABS(BTA).GT.ABS(VAL)) GO TO 902
      IF(J.EQ.4) GO TO 903
      ZZ=ZJ
900   ARCM=ARC
902   NWF=N
      ZCEN1=0.
      GCORE1=0.
      CORE1=0.
      DO 904 I=II,2,-2
      GCORE1=GCORE1+GWAKE(I)
      ZCEN1=ZCEN1+GWAKE(I)*ZWAKE(I)
      CORE1=CORE1+GWAKE(I)*WCORE(I)**3
904   CONTINUE
      GWAKE(2)=GCORE1
      ZWAKE(2)=ZCEN1/GCORE1
      WCORE(2)=(CORE1/GCORE1)**(1./3.)
C     ------------------------------------------------------------

C     REARRANGE THE INDICES OF THE VORTICES--------------------
      NI=(NWAKEF-NWF)*2
      IWF=NWF*2
      DO 905 JJ=4,IWF,2
      ZC1=ZWAKE(JJ+NI)
```

```
      GC1=GWAKE(JJ+NI)
      WC1=WCORE(JJ+NI)
      ZWAKE(JJ)=ZC1
      GWAKE(JJ)=GC1
905   WCORE(JJ)=WC1
      GO TO 906
903   NWF=NWAKEF
C     ------------------------------------------------------------

906   RETURN
      END


      SUBROUTINE PRESSURE(ISTEP)

      IMPLICIT COMPLEX Z

      COMMON/OPTION/INSTEP,NSTEP,NPLT,IFPL,IFFL,UMAX
      COMMON/SEPRAT/ISEP(2),ZSSP(2)
      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,DELTA
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)
      COMMON/PRESSURE/CPL(150),CPU(150),CPML(150),CPMU(150)
      COMMON/SVEL/XUL(150),XUU(150),XVL(150),XVU(150),
     *                              XWL(150),XWU(150)

      DO 100 I=1,NBODYH
        KK=I
        IF(I.EQ.ISEP) GO TO 100
        CALL CRSVEL(1,KK,VL,VU,WL,WU)
        CALL AXVEL(KK,VL,VU,WL,WU,UL,UU)
        XUL(I)=COS(ALPH)+UL
        XUU(I)=COS(ALPH)+UU
        XVL(I)=VL
        XVU(I)=VU
        XWL(I)=WL
        XWU(I)=WU
        CPL(I)=1.-(COS(ALPH)+UL)**2-VL**2-WL**2
        CPML(I)=CPML(I)+CPL(I)
        CPU(I)=1.-(COS(ALPH)+UU)**2-VU**2-WU**2
        CPMU(I)=CPMU(I)+CPU(I)
100   CONTINUE

      IF(ISTEP.NE.NSTEP) GO TO 200
      DO 300 I=1,NBODYH
        CPL(I)=CPML(I)/31.
        CPU(I)=CPMU(I)/31.
300   WRITE(60,400)CPL(I),CPU(I)
400   FORMAT(1X,E11.4,5X,E11.4)

200   RETURN
      END

C     ============================================================

      SUBROUTINE CRSVEL(IS,I,VL,VU,WL,WU)

      IMPLICIT COMPLEX Z

      COMMON/SEPRAT/ISEP(2),ZSSP(2)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)
      COMMON/MATRIX/NBC,AA(150,150),AX(150)
      COMMON/BODY/M1,M2,M3,M4

      ZI=ZBODY(I)

C     INDUCED VELOCITY BY BOUND VRTX PANELS AND WAKE    ------------
```

```
      ZVC=ZV(ZI)
C     ------------------------------------------------------------

C     TOTAL INDUCED VELOCITY    ----------------------------------
      IF(IS.EQ.1) THEN
      ZVDIS=ZSBD(I)*AX(I)
      ELSE IF(IS.EQ.2) THEN
      ZVDIS=ZSBD(I-1)*AX(I)
      ELSE IF(IS.EQ.3) THEN
      ZVDIS=ZSBD(I-1)*AX(I)
      ELSE
      ZVDIS=ZSBD(I)*AX(I)
      ENDIF
C     ------------------------------------------------------------

      RETURN
      END


C     ============================================================

      SUBROUTINE AXVEL(I,VL,VU,WL,WU,UU,UL)

C     CALCULATION OF AXIAL VELOCITY ON WING SURFACE    ------------

      IMPLICIT COMPLEX Z

      COMMON/SEPRAT/ISEP(2),ZSSP(2)
      COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,DELTA
      COMMON/MATRIX/NBC,AA(150,150),AX(150)
      COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)
      COMMON/WAKEGM/X,NWAKEL,NWAKEF,ZWAKE(150),GWAKE(150),WCORE(150)

      ZI=ZBODY(I)

C     POTENTIAL BY FREE STREAM    ---------------------------------
      PHF=SIN(ALPH)*REAL(ZBODY(I))
C     POTENTIAL BY BOUND VORTEX PANELS    -------------------------

CC     BY LOWER PANELS    --------------------------
      PHBLL=0.
      PHBLU=0.
      DO 100 J=1,NBODYH-1
        JP1=J+1
        ZMID=(ZBODY(J)+ZBODY(JP1))/2.
        AM=AIMAG(CLOG(ZI-ZMID))
        IF(I.EQ.J) GO TO 200
        A=AIMAG(CLOG(ZI-ZBODY(J)))
        IF(I.EQ.JP1) GO TO 300
        AP=AIMAG(CLOG(ZI-ZBODY(JP1)))
        GO TO 400
200     A=0.
        AP=AIMAG(CLOG(ZI-ZBODY(JP1)))
        GO TO 400
300     AP=0.
400     IF(A.GE.0.) GO TO 500
        ARG=A+2.*PI
        GO TO 600
500     ARG=A
600     IF(AM.GE.0.) GO TO 700
        ARGM=AM+2.*PI
        GO TO 800
700     ARGM=AM
800     IF(AP.GE.0.) GO TO 900
        ARGP=AP+2.*PI
        GO TO 110
```

```
900      ARGP=AP
110      ARGL=ARG
         ARGML=ARGM
         ARGPL=ARGP
         IF(I.LE.J) THEN
         ARGU=ARGL+2.*PI
         ARGMU=ARGML+2.*PI
         ARGPU=ARGPL+2.*PI
         ELSE IF(I.EQ.J+1) THEN
         ARGU=ARGL
         ARGMU=ARGML
         ARGPU=ARGPL+2.*PI
         ELSE
         ARGU=ARGL
         ARGMU=ARGML
         ARGPU=ARGPL
         ENDIF
         PHBLL=PHBLL-AX(J)*ALBD(J)*(ARGL+4.*ARGML+ARGPL)/(12.*PI)-
     *          (AX(JP1)-AX(J))*ALBD(J)*
     *                        (2.*ARGML+ARGPL)/(12.*PI)
         PHBLU=PHBLU-AX(J)*ALBD(J)*(ARGU+4.*ARGMU+ARGPU)/(12.*PI)-
     *          (AX(JP1)-AX(J))*ALBD(J)*
     *                        (2.*ARGMU+ARGPU)/(12.*PI)
100    CONTINUE
CC     ------------------------------------------

CC     BY UPPER PANELS    --------------------------
       PHBUL=0.
       PHBUU=0.

       DO 410 J=NBODY,NBODYH+1,-1
         JP1=1+MOD(J+NBODY,NBODY)
         ZMID=0.5*(ZBODY(J)+ZBODY(JP1))
         AM=AIMAG(CLOG(ZI-ZMID))
         IF(I.EQ.JP1) THEN
         AP=0.
         A=AIMAG(CLOG(ZI-ZBODY(J)))
         ELSE
         AP=AIMAG(CLOG(ZI-ZBODY(JP1)))
         A=AIMAG(CLOG(ZI-ZBODY(J)))
         ENDIF
         IF(A.LT.0.) THEN
         ARG=A+2.*PI
         ELSE
         ARG=A
         ENDIF
         IF(AM.LT.0.) THEN
         ARGM=AM+2.*PI
         ELSE
         ARGM=AM
         ENDIF
         IF(AP.LT.0.) THEN
         ARGP=AP+2.*PI
         ELSE
         ARGP=AP
         ENDIF
         ARGL=ARG
         ARGML=ARGM
         ARGPL=ARGP
         IF(I.EQ.JP1)THEN
         ARGU=ARGL+2.*PI
         ELSE
         ARGU=ARGL
         ENDIF
         ARGMU=ARGML
         ARGPU=ARGPL
```

```
        PHBUL=PHBUL-ALBD(J-1)*(AX(JP1)*ARGPL+2.*(AX(JP1)+AX(J))*ARGML
     *           +AX(J)*ARGL)/(12.*PI)
        PHBUU=PHBUU-ALBD(J-1)*(AX(JP1)*ARGPU+2.*(AX(JP1)+AX(J))*ARGMU
     *           +AX(J)*ARGU)/(12.*PI)
410    CONTINUE
       PHBL=PHBLL+PHBUL
       PHBU=PHBLU+PHBUU
CC  ------------------------------------------
C   ------------------------------------------------------------

C  POTENTIAL BY SEPARATED VORTICES   ---------------------------
       PHWL=0.
       PHWU=0.

       DO 510 J=1,NWAKE
         AL=AIMAG(CLOG(ZI-ZWAKE(J)))
         AU=AIMAG(CLOG(ZI-CONJG(ZWAKE(J))))
         IF(AL.GE.0.) GO TO 610
         ARGL=AL+2.*PI
         GO TO 710
610      ARGL=AL
710      IF(AU.GE.0.) GO TO 810
         ARGU=AU+2.*PI
         GO TO 910
810      ARGU=AU
910      PHWL=PHWL-GWAKE(J)*ARGL/(2.*PI)+GWAKE(J)*ARGU/(2.*PI)
         PHWU=PHWU-GWAKE(J)*(ARGL+2.*PI)/(2.*PI)
     *           +GWAKE(J)*ARGU/(2.*PI)
510    CONTINUE
C   ------------------------------------------

C   TOTAL POTENTIAL ON UPPER AND LOWER SURFACE   ----------------
       PHL=PHF+PHBL+PHWL
       PHU=PHF+PHBU+PHWU
C   ------------------------------------------------------------

C   AXIAL VELOCITY ON UPPER AND LOWER SURFACE   -----------------
       UU=TAN(EPSL)*(PHU-VU*REAL(ZI)-WU*AIMAG(ZI))
       UL=TAN(EPSL)*(PHL-VL*REAL(ZI)-WL*AIMAG(ZI))
C   ------------------------------------------------------------

       RETURN
       END



       SUBROUTINE LIFT

       IMPLICIT COMPLEX Z

       COMMON/BODY/M1,M2,M3,M4
       COMMON/PARAMT/PI,DX,X0,ALPH,EPSL,A0,XK,DELTA
       COMMON/GOMTRY/TGBD,NBODY,NBODYH,ZBODY(150),ZSBD(150),ALBD(150)
       COMMON/PRESSURE/CPL(150),CPU(150),CPML(150),CPMU(150)
       COMMON/LIFT/XLFTW,XLFTF,XDRGW,XDRGF,XLIFT,XDRAG,RATIO

   FORCES ON THE MAIN WING  ------------------------------

       WFU=0.
       WFL=0.
       DO 100 I=1,M1-1
         WFU=WFU+(CPU(I)+CPU(I+1))*ALBD(I)/2.
100    WFL=WFL+(CPL(I)+CPL(I+1))*ALBD(I)/2.

       XLFTW=(WFL-WFU)*COS(ALPH)
```

```
      XDRGW=(WFL-WFU)*SIN(ALPH)
C     ------------------------------------------------------
C     FORCES ON FLAP    ---------------------------------------

      FFL=0.
      FFU=0.
      DO 300 I=M1,M2-1
        FFU=FFU+(CPU(I)+CPU(I+1))*ALBD(I)/2.
300   FFL=(CPL(I)+CPL(I+1))*ALBD(I)/2.

      XLFTF=(FFL-FFU)*COS(DELTA)*(1.+SIN(EPSL)*SIN(ALPH)*TAN(DELTA))
      XDRGF=(FFL-FFU)*COS(DELTA)*SIN(ALPH)*
     *                  (1.-SIN(EPSL)*TAN(DELTA)/SIN(ALPH))
C     ------------------------------------------------------

C     TOTAL FORCES    ---------------------------------------
      XLIFT=(XLFTW+XLFTF)/XK
      XDRAG=(XDRGW+XDRGF)/XK

      RATIO=XLIFT/XDRAG

      RETURN
      END
```